CANADIAN THESES ON MICROFICHE

I.S.B.N.

THESES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL-339 (r. 82/08)

Canada

198 0-315-16106-X

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Division

Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

64015

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

LAWRENCE DOUGLAS RONEY

Date of Birth — Date de naissance

FEB 22 1943

Country of Birth — Lieu de naissance

CANADA.

Permanent Address — Résidence fixe

1031 CANTABRIAN DR. S.W.
CALGARY, ALTA
T2W 1M3

Title of Thesis — Titre de la thèse

COMPUTER LITERACY FOR JUNIOR HIGH SCHOOLS: A COURSE OF STUDIES.

University — Université

UNIVERSITY OF ALBERTA.

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MASTERS OF EDUCATION

Year this degree conferred — Année d'obtention de ce grade

1983

Name of Supervisor — Nom du directeur de thèse

DR M. W. PETRUK

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

APRIL 18, 1983

Signature

Lawrence D Roney

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

COMPUTER LITERACY IN JUNIOR HIGH SCHOOLS:
A COURSE OF STUDIES

by

©

LAWRENCE DOUGLAS RONEY

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTERS OF EDUCATION

DEPARTMENT OF INDUSTRIAL AND VOCATIONAL EDUCATION

EDMONTON, ALBERTA

SPRING, 1983

micro

THE UNIVERSITY OF ALBERTA

RELEASE FORM

LAWRENCE DOUGLAS RONEY

COMPUTER LITERACY IN JUNIOR HIGH SCHOOLS:
A COURSE OF STUDIES

MASTERS OF EDUCATION
SPRING, 1983

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive abstracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed) Lawrence D Roney

1031 Cantabrian Dr. S.W.
Calgary, Alberta
T2W 1M3

DATED .April.18.. 1983

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled COMPUTER LITERACY IN JUNIOR HIGH SCHOOLS: A COURSE OF STUDIES submitted by LAWRENCE DOUGLAS RONEY in partial fulfillment of the requirements for the degeree of MASTERS OF EDUCATION.

....

Supervisor

....

Date ..April 18, 1983

ABSTRACT

Computer literacy will become increasingly important to our society as the far reaching impact of computers is felt. Old skills will become obsolete and individuals will be forced to perform at new levels as repetitive tasks requiring low level skills will be taken over by computers or computer controlled robots. An increasing number of jobs will require direct interactive use of computers on a daily basis.

Recent advances in microelectronics and computer technology have provided computer access for personal use. The availability of microcomputers has resulted in an increased demand for computer literacy in our educational institutions.

Curriculum development and the preparation of educators to teach computer literacy is lagging behind the hardware availability. The Steering Committee, struck by Alberta Education to make recommendations for implementation of computer literacy in Alberta schools, has suggested that computer literacy programs be available in junior high schools in September, 1984. To date, no course of studies has been developed and tested to assist educators in meeting this recommendation.

This study has undertaken to develop and test a course of studies which can be taught in option time blocks in

junior high schools in Alberta. The aim was to provide a course that is compatible with the provincial guidelines in areas of content, hours of instruction and computer access, and student/computer ratio. A description of classroom organization techniques and support materials are included along with the objectives for this study. Problem areas are identified and solutions put forward.

A statistical analysis of the pretest and posttest results has indicated that the experimental group had significantly higher scores on the posttest as compare with the control group all grade levels. However, the return on time and effort invested is considerably greater at the grade eight and nine levels, indicating that the course of studies developed is more appropriate for these grades than for grade seven students. Additional comments are made in the study regarding difficulties encountered at the grade seven level.

The findings are generally supportive of the recommendations made by the Steering Committee. Computer literacy can be effectively implemented in junior high schools with fifty hours of instructions and computer access time using a student/computer ratio of three to one.

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to the many people who have assisted with this study.

In particular, recognition is due to Dr. M.W. Petruk who so capably served a mentor throughout this study.

Also recognized is the contribution of the other members of the thesis committee, Dr. F. Ilott and Dr. D. Sawada.

My thanks are extended to the following people from Vincent Massey Junior High School:

- Mr. L. Sampson, Principal, for his support of the Computer Literacy program in the school.
- Ms. D. Fife, Mr. A. Penner and Mrs. K. Vanderstaam who taught the course of studies.
- Mrs. M. Neiderweiser and Ms. J. McAllister for typing and proofreading the manuscript.
- the rest of the staff and students who were so tolerant and supportive throughout.

Appreciation is also expented to Mr. J. Curda, Mr. B. Kemp and Mrs. T. Kemp who took the time to assess the evaluation document for validity.

I would also like to thank Mr. R. Baker and Dr. A. Morris for the initial equipment and support which initiated this study.

Finally, to my family, who have sacrificed so much to support me in this study, a special 'thank you'.

TABLE OF CONTENTS

Page

CHAPTER I 1

Statement of the problem 3

Hypothesis 4

Significance of the Study 5

Scope of the Study 5

Limitations 5

Operational Definitions 6

CHAPTER II 8

Justification for Computer Literacy 8

What is Computer Literacy? 11

What Constitutes a Computer Literacy Program of Studies? 14

Content and Presentation of Computer Literacy 18

Obstacles to Implementing Computer Literacy in Schools 20

Summary 22

CHAPTER III 24

Developmental Phase 25

Revised Program of Studies 28

Program Expansion and Evaluation 31

Formal Evaluation of the Course of Studies 33

CHAPTER IV 39

Analysis of the Data 40

CHAPTER V 49

Summary and Conclusions 49

Discussion 49

Recommendations for Further Study and Action 52

BIBLIOGRAPHY 56

APPENDIX A Course Objectives 59

APPENDIX B The Course of Studies 68

APPENDIX C The Pretest 75

APPENDIX D The Posttest 83

LIST OF TABLES

Table 1. Pretest Data Analysis .... Page 41

Table 2. Mean Scores by Group for Three Grade Levels.................... Page 43

Table 3 Analysis of Variance on Posttest .... Page 44

LIST OF FIGURES

Figure 1. Mean Gain Scores for
Experimental Groups .... Page 46

CHAPTER I

A major technological breakthrough occurred in 1970 with the development of the microprocessor integrated circuit or "computer on a chip". This tiny chip has revolutionized the electronics and communications industries. Computers and computer controlled devices based on the microprocessor chip are appearing in many segments of business and industry as well as in private homes. Repetitive tasks requiring lower level skills are being performed by computers or computer controlled robots. Few segments of our society have escaped the impact of "computerization".

Advances in microelectronic technology have resulted in entire industries being phased out while new industries have been born. The job requirements for individuals employed in an evolving marketplace have also changed. The mechanical watch industry has been virtually replaced by microprocessor-based digital watches. New video games based on this same technology have become a new and rapidly expanding industry. Engineers, architects and scientists now use computer simulations to test their ideas. Office staff frequently find computers and word processors replacing the typewriters and filing cabinets they used to work with. Managers are frequently required to interact directly with computers in budgetting, forecasting and modelling to assist with managment decisions.

The introduction of the microcomputer in 1975 effectively expanded access to computing beyond large companies and highly trained specialists. Reductions in size and cost, coupled with the development of higher level languages have placed the power of computers within reach of an ever increasing segment of our population.

The rate of change of this technology and its impact on a wide range of segments of our society has resulted in the need for change within educational institutions. Previously, only a few specialists were required by large corporations to operate their computers and university courses provided the preparation needed for these positions. It is now becoming increasingly important that professional people, business managers, office staff and private individuals have an understanding of the role that computers play in their lives. Formal educational institutions will be charged with the responsibility for providing their graduates with the knowledge required to meet the challenge resulting from changes brought about by the impact of computers on society.

While the need for systems analysts and computer programming specialists will continue, a need for another level of computer knowledge, commonly referred to as "computer literacy" is emerging. The level of computer literacy required by individuals will vary with the role they assume in society. However, a general knowledge of how to operate computers and their applications and implications in our society is becoming increasingly important to all members of a highly technical, information-based society.

It is this general level of computer literacy that the pre-college educational institutions will be expected to address. These institutions must accept the responsibility for preparing their graduates for the role they are to assume in society. Schools that continue to graduate students without at least a general level of computer literacy are not meeting their responsibility and can expect pressure from the business sector, the private sector and ultimately the government in the form of legislation making computer literacy a compulsory component of formal education.

Statement of the Problem.

The problem posed for this study is:

Can computer literacy be effectively taught in junior high schools in Alberta within existing option time blocks?

A number of considerations were taken into account in posing this problem:

1. Computer literacy does not "logically belong" within the domain of any existing core curriculum but rather encompasses all of them.
2. The insertion of computer literacy into a core

subject curriculum would result in undesirable reductions in content and/or time for the existing subject matter.

3. While computer literacy is highly desirable for all students, it was felt that it should not be imposed on learners at the expense of other educational experiences, but rather should be offered as an option at the junior high school level.
4. The use of option time blocks creates a minimum of disruption in school organization and provides maximum flexibility in providing instructors for computer literacy classes from a number of departments within the school.

The underlying concern in answering the question posed by the problem is the feasibility of effectively teaching computer literacy within the existing organizational structure without detracting from existing educational experiences being offered to junior high school students.

Hypothesis.

The course of studies developed for this study will be taught to junior high school students as the treatment in a pretest-posttest equivalent-group experimental design to test the following null hypotheses:

1. There will be no signifigant difference in pretest mean scores between the experimental and control groups.

2. Following the treatment of the experimental group, there will be no significant difference between the mean of the experimental group and the control group when the pretest and posttest scores are compared.

Significance of this Study

This study is significant because, at this point in time, no course of studies in computer literacy in junior high schools has been tested in Alberta and this study will test such a course of studies. Development of a curriculum for computer literacy in junior high schools is under way at the provincial level so the findings of this study will provide additional information to support this effort.

Scope of the Study.

1. This study is limited in scope to dealing with the knowledge required for computer literacy. It is recognized that skills and attitudes also are important to computer literate individuals. Knowledge was chosen as the focus of this study in the belief that it is the foundation on which skills and attitudes can be built.

Limitations.

1. Testing of the program was limited by the lack of

junior high schools with the necessary facilities and available teachers who had sufficient computer literacy training to enable them to present the course of studies.

2. The selection of subjects and instructors to participate in this study was limited by the organizational structure of the junior high school in which the study was conducted.

Operational Definitions

1. Computer literacy - is the knowledge about computers and their implications for society that a layperson needs to function effectively as a productive member of society.

2. Course of Studies - The set of objectives, content and strategies for organization and teaching which are appropriate for junior high students in Alberta.

3. Junior High Schools - Educational institutions which accomodate student in grades seven, eight, and nine. Student usually range in age from twelve to fifteen years.

4. Hardware - The electronic equipment associated with computing including the computer, video display screen, disk drive, tape recorder, and printer.

5. Software - The programs or magnetically stored

operating instructions for use in computers.

6. Courseware - The printed and audio-visual materials associated with the presentation of information about computer literacy.

7. Computer - A high speed information processing device which operates under a prepared set of instructions to complete a task.

CHAPTER II

Since the purpose of this study was to develop a course of studies in Computer Literacy for junior high school students and evaluate its effectiveness, a review of the literature was undertaken with the following concerns in mind:

1. The justification for including Computer Literacy in formal education.
2. Definitions of Computer Literacy.
3. What should be included in a Computer Literacy course of studies and how should it be presented in the schools?
4. What obstacles are likely to be encountered in implementing Computer Literacy courses in the schools?

Justification

Fifteen years ago, Michael (1968) was expressing the view that cybernetics or the applications of computers and the associated automation was going to have a very significant impact on society. While it was generally recognized that blue collar workers were going to be replaced on assembly lines, he warned that many skilled white collar workers would find their jobs changed or non-existant. Accountants would be replaced by multipurpose computers. Engineers and office workers would find major changes in the tasks that they would be expected to carry out. The customary expectation of continuity from school to career will need to be replaced by the expectation of a cycle of training and work as the impact of technology changes the workplace. Ignorance of the capabilities of computers and their impact on society will leave the members of society functionally illiterate.

Prior to the development of the integrated circuit or "chip" in the early nineteen seventies, very little had been done to actively implement computer literacy in the schools. Access to computers was limited because of the size and expense of computer hardware. The introduction of the microcomputer or personal computer in 1975 made it possible for a much larger segment of the population to have access to computers. As a result, the computer had an increased impact on our daily lives and educators were beginning to face increasing pressure to prepare students for the changes that the computer would bring.

In the literature, there is general agreement that a computer literate population is a desirable goal for which to strive. Recent advances in microelectronics and computer technology has resulted in major changes in industry. Osborne (1979) predicted that half of the jobs that exist in the industrial world will be eliminated in the next twenty five years and that the remaining jobs will require new skills. Entire industries will be crippled or wiped out in the same manner that the mechanical watch industry has been destroyed by low priced digital watches based on the integrated circuit. The workforce of tomorrow must be prepared to contend with an increased rate of change. In a presentation on learning in the eighties, Forsythe and Hart (1980) stated that "the difference between the impact that computers will have from that of other major technologies is because the time scale of their introduction on a massive scale is much less than that of a human generation" (pp.471-72).

Statland (1979) concurs that the rate of change in the computer industry is such that its impact on society is a major challenge. He reports that:

> In the 1970's it was estimated that one out of six men, women and children had their daily lives affected by the computer. This statistic should change to one out of every two lives that will be affected on a daily basis by 1984. (p.16)

Much of the increased impact will be due to the fact that we are shifting from being an industrial society to being an information based society. The computer, with its ability to handle massive amounts of information very rapidly, is the driving force behind the change. According to Molnar (1978):

> Information has become a national commodity and a national resource; it has altered the very nature of work. We, as a nation, have moved from being predominantly an industrial society to being an information society. .... In an information society

> a computer literate populace is as important as energy and raw materials are to an industrial society. Computer literacy is a prerequesite to effective participation in an information society and as much a social obligation as reading literacy. (pp.31-2)

The importance of a computer literate populace was also stressed by Luehrmann (1980). In his opinion:

> Computing plays such a crucial role in everyday life and in the technological future of this nation that the general public's ignorance of this subject constitutes a national crisis.
>
> The ability to use computers is as basic and necessary to a person's formal education as reading, writing, and arithmetic. As jobs become increasinly oriented toward the use of information, society demands and rewards individuals who know how to use information systems. (p.98)

The responsibility for the social obligation of computer literacy will fall to the schools. Anderson (1980) feels that this is logical since the major responsibilty for preparing citizens to assume their role in society resides in the precollege educational institutions.

The need for computer literacy has been identified by a number of authors and there is general agreement that action should be take immediately if educators are to prepare the students of today for their role in a rapidly changing society.

What is Computer Literacy?

While there is a general concensus in the literature that computer literacy should be part of the formal education of all students, there is no such concensus on what computer literacy is. Initially, computer literacy was understood to mean a level of understanding which would enable the individual to talk about computers. Little or no experience working with computers was implied. In a program of this type, students were exposed to movies and talks about computers, toured computer centers and discussed how computers were used in business, government, and science. This level of understanding is now generally regarded as computer awareness.

These concepts of computer awareness are included in a a definition of computer literacy but now computer literacy generally includes more direct involvement with computers. In dicussing the Computer Literacy Project at MECC, Anderson (1980) states that:

> The project is based on the premise that computer literacy consists of a broad understanding of the role of computers in society as well as the ability to communicate with them. (p.211)

This statement about computer literacy does not consider what involvement the learner is to have with the computers. Brumbaugh (1980) was more specific. In his view:

> To be computer literate, one must be able to define, demonstrate, and/or discuss:
>
> . How computers are used
> . How computers do their work
> . How computers are programmed

> . How to use a computer
> . How computers affect our society. (p.234)

In dicussing the findings of a survey conducted by the Human Resources Research Organization, Wexler (1979) notes that computer literacy was defined as:

> Understanding the impact of computers on society, how computers work within a discipline and the actual programming process. (p.42)

Due to rapid advances in computer technology and its applications in our society, any educational program which prepares students for their evloving role in that society must adapt in response to these changes. change. Watt (1980), in discussing computer literacy, states:

> I think of computer literacy as a cultural phenomenon which includes the full range of skills, knowledge, understanding, values and relationships necessary to function effectively and comfortably as a citizen of a computer based society. The computer literacy needs of any individual will vary according to that person's particular involvement with computers, but the computer literacy requirements for the average person should expand dramatically in the next decade. (p.57)

Moursund (1981) bases his approach to computer literacy on the idea that computers will have a personal impact on students and that students will be self motivated to become computer literate because of the benefits derived from the personal use of computers. He concludes that:

> Computer literacy is a functional knowledge of computers and their effect on students and on the rest of our society. This knowledge should be at a level compatible with other knowledge and skills a student is acquiring at school. It is a knowledge based on

> understanding how computers can help us learn, how computers can help us solve problems, what computer knowledge is essential to a modern understanding of other academic areas, what is included in the field of computer and information science, computers as entertainment and what role computers will play in our changing world. This approach to computer literacy changes easily as computers become more readily available and easier to use, as we learn more about computers and integrate this knowledge into the curriculum and as the use of computers become more commonplace in homes, business, government and schools. (p.21)

The Computer Literacy Report and Recommendations (1982) submitted to Alberta Education also recognized that computer literacy will continue to evolve and states that:

> A definition of computer literacy should remain flexible and dynamic as the specific skills, knowledge and values required to be computer literate will vary with time and with the student's level of computer expertise. (p.1)

What Constitutes a Computer Literacy Program of Studies?

Despite the fact that computer literacy lacks a clear-cut definition, there is general agreement on many components that should be included. The most contentious issue is how much, if any, programming should be included.

Gawronski (1981) identifies themes that should be included in a computer literacy course including:

> 1. What a computer can and cannot do
> 2. What a program can and cannot do
> 3. How to program.
>
> Students must acquire skills in identifying and evaluating the advantages and disadvantages of using computers for selected applications. Not

> all problems are best solved by a computer. (p.613)

This description makes no mention of the influence of computers on society. The previously mentioned definition of computer literacy by Brumbach (1980) includes how computers are programmed as well as how they effect our society but goes on the raise the issue of whether the writing of programs should be a requirement. He does suggest that learners should know how computers are programmed and stresses that, at the very least, "hands-on" experience in using computers should be a requirement.

Johnson (1980) proposed a program which included the previously mentioned topics but placed more emphasis on the advances of technology, social implications and futuristics including trends in artificial intelligence and robotics. Her recommendations for a computer literacy course include:

> No course prerequisite, low level technical introductory computer concepts, three-fourths of content on nonprogramming materials and one-fourth on computer programming - only if a computer is available. (p.20)

A rather detailed set of recommendations for the content of computer literacy was presented by Johnson, Anderson, Hansen and Klassen (1980). The topics and objectives were grouped under the headings of Hardware, Programming and Algorithms, Software and Data processing, Applications, Impact, and Attitudes, Values and Motivation.

The authors recognize that the coverage of the full scope of what has been outlined is not feasible within any single course and thus it is intended as a guideline for a long-term program. They see educators selecting areas and activities that are appropriate for the present situation while keeping in mind the overall objectives so that the range of topics can be covered in the student's elementary and secondary education.

In Alberta, the content, goals and objectives proposed in the Computer Literacy Report and Recommendations (1982) suggest that the junior high school course include the following topics:

> Topic 1: What is a Computer
> Topic 2: How to Operate a Computer
> Topic 3: Computer Programming
> Topic 4: Computer Uses in Society
> Topic 5: How Computers Affect Society. (p.5)

The report recommends that the program should concentrate on Topic 3 to develop a working knowledge of computers. The allocation of time will vary depending on the previous exposure that students have had to computers. As the Elementary computer literacy unit is implemented to cover the content of Topics 1 and 2, it is suggested that sixty to eighty percent of the time be spent on Topic 3 with about one third of the course time involving "hands-on" activities. To achieve this level of computer access time, the report recommends a ratio of one computer for every three students.

While this amount of time spent on computer progaramming is considerable, there is strong support for the inclusion of programming skills in computer literacy. As Klitzner (1981) sees it, programming is:

> Probably the best way to appreciate how dumb and and how smart these machines are, to appreciate how detailed and precise instructions have to be, and to see how errors can creep into the process. (p.64)

Further support is lent to the inclusion of programming in computer literacy by Gawronski (1981) in his statement that:

> A computer-literate individual should know how to write an original program. The computer is a problem-solving tool and writing computer programs utilizes problem-solving abilities. (p.614)

Moursund(1981) is supportive of including programming and problem solving as part of his approach to computer literacy. Being able to write and use programs will add to the personal benefit that the individual will derive from being computer literate. However, he expands the definition of computer programming to include learning to use special purpose programs like word processing and statistical analysis. He contends that learning to use a special purpose language applies all the same principles as learning a general purpose language like BASIC or Pascal and the user is clearly directing the system.

In his opinion, there is a trend to the production of special purpose software systems for information handling and that learning to use them effectively is comparable in difficulty to learning a programming language. In concluding, he states that:

> Including the use of information retrieval, word processing, statistics and other application systems in our definition of programming greatly broadens the scope of what it means to learn to program. Learning to use applications systems is a rapidly growing part of computer programming; it will eventually become the dominant part. All students can learn to use applications systems, and this is an essential part of becomming computer literate. (p.15)

In short, a meaningful course of studies in computer literacy must continue to evolve in response to changes in computers, the software available and the growing understanding of their applications in society.

Content and Presentation of Computer Literacy

While a number of possible methods of presentation have been suggested, the final choice will be determined by the facilities and teaching materials available.

A summary of some possible methods of implementation is presented by Klitzner (1981). One method would be to set up the computer as the instructor. For this method to be effective, good quality software must be available and sufficient hardware to allow each student reasonable computer access time.

A second alternative is to use the more traditional approach using teachers, texts, pencil and paper. According to Klitzner (1981), this is the method favored by Luehrmann since print based materials are straightforward and learners know who is in charge when they sit down at the computer.

A mix of the first two methods would be the method that Klitzner says would be favored by Moursund and Billings. In this method, there would be a balance of computer instruction, print based instruction and teacher directed lessons. The method employed would be determined by the topic being covered as well as the available facility. It is doubtful that, in the near future, junior high schools will have sufficient hardware to use exclusively computer based instruction even with good software available.

Leong (1975) investigated three approaches to teaching computer programmming to junior high school students. These methods included batch processing using punched cards which were processed outside class time, an interactive approach in which students were working on terminals during class, and a simulation approach with no computer access. In the simulation method students learned to "act out" the execution of programs in class via "conceptual computers".

The results of a programming skills test and questionaire indicated that there was no significant difference in the level of programming skills attained with the three teaching methods. There was a noticeable difference, however, in the interest level. The author reported a sustained higher interest level when the learner had an opportunity to interact with the computer directly via a terminal.

In one of the few references to functioning with limited hardware and large classes, Piele (1979) discussed assigning two students to one machine. He stated that:

> The original purpose for doubling up was to provide more computer time each week. But it turned out to be valuable for a completely different reason - cooperation. The students helped each other figure out the effect of each new command. .... The computer was the focus and facilitator for cooperative problem solving. (p.133)

Obstacles to Implementing Computer Literacy in Schools.

According to Moursund (1979), Barriers to progress in making instructional use of computers can be divided into two categories. In one he places the shortage of hardware,software and courseware; the other includes lack of knowledge of computers on the part of individual teachers and school administrators. He contends that the latter:

> Is the major and continuing bottleneck. Without knowledgeable teachers and supportive administrators, progress will be painfully slow. With them, progress is rapid, even in light of inadequate hardware, software and courseware. (p.39)

This point of view is supported by Stevens (1980). In summarizing a survey of Nebraska educators in all subject areas, the author found that while over seventy percent of the teachers surveyed were supportive of the concept that high school students should be able to demonstrate an understanding of computers and their role in society over ninety percent indicated that they did not feel qualified to teach computer literacy. He also found that the attitudes of teachers presented an obstacle to the implementation of computers because they felt threatened by a loss of security and classroom authority. Therefore a computer literacy component in teacher preparation was deemed to be a first step in implementing computer literacy.

Summarizing factors that will effect the rate of adoption of computers in education, Solntseff (1981) stated:

> 1) Computers have to be perceived to offer advantages over the traditional manner of performing a given task; 2) the use of computers has to be seen to be no more complicated than that of traditional tools; 3) computers have to be easily accessible so that prospective users can experiment with them; 4) the use of computers has to be easily observable by persons contemplating their use; 5) the values, experiences, and needs of teachers and administrators should be "in tune" with the integration of computers into the system. (p.100)

According to Luehrmann (1980), four specific needs have to be met before computer literacy can be achieved:

. adequate and appropriate equipment in every secondary school
. an available , usable curriculum with materials for students and teachers
. one or more teachers in each school trained in teaching computer use
. community, political, and financial support for school-based programs. (p.98)

He contends that all of these needs can be readily met if society decides that computer literacy is a high priority in our education system.

Summary.

In the foreseeable future, the definition of computer literacy will continue to evolve. A number of factors will be responsible for these changes:

1. Further advances will be made in computer technology so that equipment will be more avalialbe and easier to use.
2. Software and courseware will improve, making it feasible to offer improved courses to the students.
3. Students will become more sophisticated in their knowledge of computers.
4. Teachers will become more knowledgeable about computers and how they can be used in education as both a teaching tool and an object of study.

The course of studies will reflect an awareness of what computers are and how to operate them, a knowledge of the elements of computer programming, and an understanding of the impact of computers on society. Considerable developmental work will be needed to identify effective and efficient method of achieving the goals of computer literacy within the limitations imposed by budgets, facilities, available materials, and the expertise of educators.

The rate of adoption of computers in education will, in all liklihood, be determined by pressures brought to bear from a "grass roots" demand that student aquire computer literacy in school. Legislation in response to this demand may make computer literacy compulsory and thus force local school jurisdictions to place computer literacy high on their spending priorities.

Preparation of teachers to present this program will of necessity be a shared responsibility. The individual teacher must take the initiative to become aware of computers in education. Inservices courses offered by school boards and educational institutions will be necessary to upgrade the skills of teachers presently working in the system and teacher training institutions will be responsible for developing and delivering adequate training for the teachers of the future.

The preparation and evaluation of courseware will present a challenge to educators especially in light of the evolving nature of computer literacy. Incividual teachers will prepare some of the needed materials but, for the most part, will lack the time, the expertise and the resources to develop materials quickly enough to meet the demand. Support from governments, publishing houses, computer companies and local school jurisdictions will be necessary to meet the challenge of providing good courseware to assist in the implementation of computer literacy in the schools.

CHAPTER III

The literature clearly indicates that a computer literate population is desirable and that the responsibility for presenting this knowledge will be assigned to formal educational institiutions. If public school systems are to meet this challenge, then teachers will of necessity become computer literate and courseware must be prepared to enable them to implement meaningful computer literacy programs.

This chapter will summarize the development, implementation and evaluation of a computer literacy program of studies for junior high school students.

The first task was to gather information and ideas as to what should be included in a computer literacy program and how it might be implemented within an existing school program. The literature indicated a diversity of opinion on what computer literacy is and what should be included in a program of studies which would result in computer literate citizens. Very little has been written on the actual administration of a program to provide classroom teachers with strategies for achieving the objectives set out for the course of studies.

A synthesis of the material gathered resulted in a preliminary outline with objectives being stated and trial strategies set out. The original emphasis was on a "hands on" approach to learning about computers and how they can be programmed. As this approach is highly motivating, direct use of the computer should remain an integral part of any program of studies in computer literacy.

Developmental Phase.

The course of studies was developed with grade nine students over a period of two years with formative evaluation being undertaken at various stages. A continual process of assessment and modification was conducted and changes to both the content and delivery method were made as the need for such changes was identified. Originally, the content was almost exclusively computer programming and contained more material than the students could learn in the time available. The amount of programming was reduced and other areas of computer literacy were added after the first semester and the program was presented again. This process was repeated for three semesters.

The course of studies was presented to students in a thirty-five hour time block as an option in a junior high-school in Calgary. The participants were twelve grade nine students who had expressed an interest in learning about computers. The available equipment included two Apple II Plus computers, one disk drive, one tape recorder, two video display screens and one printer. It was set up in a small classroom complete with blackboard, overhead projector and screen. This room was designated as a computer laboratory so the equipment could be left set up.

A number of difficulties were identified and addressed during this first option.

1. There was a shortage of support material which could be used with the program. Manuals were too technical or organized in such a way that they were difficult for students to use.

2. With a ratio of six students to one computer, students had difficulty getting sufficient computer access time.

3. The computer time allocated to students was not being used efficiently.

To alleviate the first problem, printed materials were purchased as they were identified and additional materials were authored. Audio-visual materials were obtained to further support the concepts being presented. These materials are included in Appendix D.

Extra computer time before school,at noon hour and after school was allocated to make it possible for students to complete assigned tasks. These supervised time blocks were used very heavily throughout the program.

Two factors were identified as contributing to the problem of inefficient use of computer time. Students tended to author programs directly on the computer and an excessive amount of time was spent revising poorly planned programs. The solution to this problem was to have students write out all programs and have them checked by the instructor before they were assigned computer access time.

The second factor contributing to inefficient use of computer time was the lack of keyboarding skills. Students were unfamiliar with a keyboard and had difficulty locating the keys required to commumicate with the computer. Business Educators in the high schools were expressing concerns that some of the habits picked up by students on the computer keyboard were making it difficult for them to succeed in typing classes. To address these concerns, a keyboarding component was added to the program. The primary focus was to introduce "touch-typing" techniques including proper posture, finger placement and keystroking. The intent was to introduce keyboarding in such a manner that the habits learned were supportive of the high school programs while at the same time having students become familiar with key placement on the keyboard.

An assessment of the program of studies was undertaken at the end of each semester. This assesment included both a formal written test and an informal discussion of the strengths and weaknesses of the program. The written evaluation indicated a need to place more emphasis on some topics and to made modifications to others where concepts had not been grasped. The informal discussions indicated that, generally, students had enjoyed the course but felt that they needed as much computer access time as possible to gain maximum benifit from the program. They felt that the extra time outside regular class time was essential. A number of students suggested that they did not feel the program was viable without the allocation of extra time outside of regular school time.

The Revised Program of Studies.

A final revision of the course of studies was undertaken prior to a final presentation and evaluation for this study. The release of the Computer Literacy Report and Recommendations submitted to Alberta Education by the Computer Literacy Steering Committee influenced the course of studies in that minor reorganization of the materials was undertaken to provide compatibility with their recommended format. The revised scope and sequence of the course of studies is outlined in the objectives presented in Appendix A.

Changes in content were minor compared with the restructuring that was undertaken in areas of administration and delivery. The priority was to make most effective use of limited computer hardware to provide students with as much meaningful "hands-on" time as possible.

The delivery methods used included large group presentations, two group rotation and three group rotation.

The choice of method depended on the material being presented. The following pattern was used in presenting the course of studies.

The introductory sessions were large group presentations where an overview of the program was given, expectations were outlined and students were paired up in preparation for future grouping. This required one to two class periods.

A two group rotation was then used with one group involved in an introduction to keyboarding while the other began an introduction to computers. The instructor worked directly with the keyboarding group while the other group worked independently on assigned tasks covering Topics One, Four and Five from the outline of objectives. Print material, video tapes and sound filmstrips were used by this group to seek out information required to complete assigned tasks.

The two groups alternated assignments after two class periods. This rotation was continued until the instructor felt that the students were able to function independantly using typing manuals to reinforce the keyboarding skills. For grade nine students, this point was reached after about eight class periods. Younger students generally required one or two extra periods.

A three group rotation was implemented at this point.

The group activities for this rotation included keyboarding, computer access and theory. The keyboarding group continued to practise on the typewriters with minimum supervision. The theory group was introduced to the material in Topics Two and Three from the outline of objective through lecture/ demonstration sessions. Students were introduced to the Apple computer and shown how to load and run programs, initialize their own disks and begin to write their own programs.

The computer access group used their class time to become familiar with the computer through a "hands-on" approach. Students developed skills in using system commands like LOAD, RUN, SAVE and LIST. In addition, they used this time to enter, run and debug programs they had written in the theory sessions.

The groups were rotated after two classes at the same station with the order of rotation being from theory to computer access to keyboarding. This three group rotation continued until the keyboarding group had covered the alphabetic keyboard. No futher class time for keyboarding was assigned but students were encouraged to practise to maintain skill levels.

The class was reorganized to a two group rotation when keyboarding was discontinued. The groups alternated between theory and computer access after two class periods. This pattern of rotation was continued for the balance of the semester.

It was now necessary to assign computer access time other than regular class time so that students could complete assigned tasks. Each student was assigned two time slots of approximately forty-five minutes per week. Where conflicts occurred with other school activities, adjustments were made. Students were also encouraged to come in whenever the computer laboratory was open to use the typewriters.

During class time there were students in the computer access group who did not have machine access. This was handled in a number of ways. Computer time was assigned first to those who had their programs written ready for entry and all written assignments completed. Those not meeting these criteria were expected to complete these tasks. If all groups were prepared , the computer time was was shared and waiting time was used for keyboarding, viewing audio-visual materials relating to the current topic, or sitting in on the lecture as a review of the concepts just covered.

Program Expansion and Evaluation

At the end of the third semester of evaluation and revision, it was felt that the course of studies was developed to the point that a formal evalution could be undertaken. Student demand for the program was growing rapidly. Support from administration and staff for an expansion of the program within the school was evident.

In order to expand the program, it was necessary to obtain more hardware and prepare staff members to present the program. Two more computers with disk drives and video monitors were purchased and one large television screen to be used for group demonstrations was obtained.

An inservice program was offered within the school and fourteen staff members from Vincent Massey Junior High School enrolled. The program consisted of thirty hours of instruction and computer access time. It was presented to the staff two evening a week for two hours per session. The material presented to the teachers was primarily the same as that offered to the students. Based on the assumption that students and teachers were starting on the same level in computer literacy it was thought that working through the program would help teachers identify potential problem areas within the course of studies.

Teachers found that they also needed extra computer access time to reinforce concepts and complete assigned tasks. They were encouraged to take computers home or arrange their time to use the computers during the day when the machines they were not required for student use. The participants were enthusiastic and often expanded on assignments that were given to reinforce the basic concepts. As with the students, it was noted that teachers with no previous typing experience required considerably more computer access time to complete assigned tasks.

Formal Evaluation Of the Course of Studies.

The following considerations influenced the formal evaluation of the course of studies:

1. The need to fit into the existing structure of Vincent Massey Junior High.

2. The availability of instructors who were interested in teaching the program and who had taken the inservice course offered.

3. The number of students to be enrolled and the selection of these students.

In order to offer the course of studies with a minimum of disruption to the school, the program was presented in option time blocks. The school was operating on a six day cycle with time blocks of forty-two minutes. The Computer Literacy Options were offered as semestered options with the class meeting three times in a six day cycle. With thirty-five hours of class time plus the additional assigned time, the students received approximately fifty hours which is the time recommended by the provincial guidelines.

Instructors were selected on the basis of expressed interest and their availabilty to teach in the option time blocks as determined by the master timetable of teaching assignments. Four instructors were assigned to teach computer literacy options for this study. The author taught one grade nine option, a Science teacher taught one grade eight and one grade nine option, a Social Studies teacher taught a grade eight option and a Mathematics teacher taught the grade seven option.

There were twelve students assigned to each of the five options so that the recommended student/computer ratio of three to one could be met. The decision to offer two options in grade eight and nine and only one in grade seven was dictated by the constraints emposed by the organizational structure of the school.

The selection of students to participate in the study was, of necessity, based on their interest in taking part since it was an option time block and they were free to choose from a wide range of options. In all cases the option was over-subscribed and a selection process was carried out.

The applicants for each class were listed and assigned consecutive numbers. The computer random number generator was used to generate a series of numbers from one to the maximum number of applicants. The first twelve students who had their number come up were included in the experimental group and the next twelve became the control group with the understanding that they would form the class for the second semester. Initially, there were one hundred and twenty participants in the study but attrition had reduced this number to one hundred and twelve by the end of the study.

The experimental design chosen for the evaluation of this study was the pretest-posttest equivalent group design. Each class consisted of an equal number of randomly selected student who had expressed an interest in the course. A pretest was administered to each member of the experimental and control groups at the beginning of the semester. The control group received no treatment while the experimental group were presented the course of studies. At the conclusion of the semester, a posttest was administered to the participants in both the experimental and control groups.

In an attempt to insure that the tests were valid and equivalent, the following procedure was carried out. A bank of proposed test questions was prepared by the instructional team. A copy of the objectives of the course of studies along with corresponding test questions was forwarded to three professional educators. They were asked to assess each question for validity. The evaluators chosen included an elementary teacher who was involved in developing and presenting computer literacy inservice courses, a junior high teacher who was teaching computer literacy and a consultant to a computer literacy project being conducted in elementary schools by the Calgary Board of Education.

Based on the feedback received from these people, a final test item bank was prepared. Test items were arbitrarily assigned to pretest and posttest until two tests were prepared that were worth one hundred marks each. In many cases, the same questions appear on both tests. Because of the procedure used and the expertise of the evaluators, the two tests are deemed to be valid and equivalent.

The course of studies was presented to the option classes in a manner similar to that used the preceding semester (as described earlier). While group structuring and order of presentation were suggested, the professionalism of the instructors was recognized and they were encouraged to tailor their delivery to their teaching styles.

In all cases, keyboarding techniques were presented and emphasized early in the course. Students were encouraged to use eight finger "touch typing" techniques throughout. As students became familiar with the keyboard, less class time was spent on keyboarding. It was noted that the grade seven students generally required more keyboarding time to reach the same proficiency as the older students.

Varying degrees of frustration were expressed by instructors and students in the early stages of the course as the result of the multiple activity grouping. Students were having to wait for the instructor to finish with another group before they could obtain the assistance they required. The effectiveness of the multiple activity grouping is dependent on the cooperation of the students and their willingness to work independently or accept assistance from fellow students when the instructor is unavailable. The maturity level of the grade nine students resulted in less frustration than that experienced by the younger students. This was especially evident in the early weeks of the course as routines and expectations were being established.

The content and sequence of the material presented is outlined in the objectives in Appendix A. All instructors followed the course of studies very closely while adjusting their presentation to suit their teaching styles and their assessment of the abilities of the students they were instructing.

The reliability of the scores on the pretest and posttest was a concern since the questions required written answers and the subjective evaluation of the responses may bring reliability into question.. To counteract this, a key of anticipated responses was prepared by the team of instructors and each test was evaluated independently by at least two markers. The scores were averaged to arrive at the final test score. The professional judgment of the combined efforts of the instructional team was deemed to be a reliable assessment of student performance. All averages were rounded to the nearest whole number. An evaluation summary form was used by instructors to summarize test results so that any major discrepencies in scores for an individual question could be seen at a glance and a re-evaluation undertaken.

A comparison of the test scores for the experimental and control groups was undertaken and the results are summarized in the next chapter.

CHAPTER IV

The course of studies prepared for this study was developed over a period of two years and formative evaluation was used at various stages of the development to assess the course. Written tests and informal feedback from students were used as assessment tools to identify areas requiring improvement.

The content of the course was influenced by the information gathered through a search of related literature. The recommendations contained in the Report and Recommendations for Computer Literacy submitted to Alberta Education were incorporated as much as possible so that the course of studies would be compatible with other developmental efforts being undertaken throughout the province. The course, as it was taught during the final evaluation, meets the provincial guidelines of fifty hours for the program with a student to computer ratio of three to one.

During final evaluation, the course of studies was taught to students in grades seven, eight and nine by four teachers. The evaluation was carried out in a junior high school in Calgary while being offered during option blocks commonly referred to in Alberta curriculum as "group B" options. Students who signed up for Computer Literacy Options were randomly assigned to the experimental and control groups. The control group participated in the study on the understanding that they would be enrolled in the option in the following semester.

A pretest was administered to the experimental and control groups at the beginning of the semester. The experimental groups were taught the experimental course of studies while the control groups were taught other unrelated options. At the end of the semester a posttest was written by each group.

The tests were scored by the instructional team consisting of the four teachers who had taught the experimental course. The data was organized in a form compatible with the requirements of statistical analysis programs available on the Amdahl 470 computer at the University of Alberta, Edmonton.

Analysis of the Data.

The pretest data gathered from the experimental and control groups was analysed to test the first null hypothesis which stated that there would be no significant difference between the mean scores of the experimental and control groups at the beginning of the study. The level of significance used for the rejection of the null hypothesis was set at the 0.05 level as is common practice in educational and psychological research. The results of the computer analysis of the pretest data is summarized in Table 1.

TABLE 1

Pretest Data Analysis

Group	Exper.		Control		T-value	Probability
	N	Mean	N	Mean		
All Subjects	56	12.78	56	8.70	2.71	0.008
Grade Nine	23	14.83	23	13.17	0.61	0.545
Grade Eight	22	12.86	21	5.10	4.08	0.000
Grade Seven	11	8.36	12	6.42	0.88	0.388

When a T-Test analysis of difference between means is carried out with a significance level of 0.05 set, a T-value which exceeds 1.96 indicates that the null hypothesis be rejected. On this basis the null hypothesis was rejected for the All Subjects group indicating that the two groups might have come from different populations and were not necessarily equivalent at the beginning of the study.

Further study of the figures in Table 1 on a grade by grade basis indicates that the null hypothesis is not rejected for the grade seven and nine groups. These experimental and control groups were likely from the same population and were not significantly different.

The grade eight T-value indicated that the difference between the experimental and control groups on the pretest was significant. The results for the grade eight groups were carefully reviewed to identify factors which may have contributed to the differences found.

The sampling method was reviewed and was found to be the same as that used for the grade seven and nines. The pretest was administered under similiar conditions for all three grade levels. The academic averages for students in the experimental and control groups in grade eight were examined to determine if a sampling error had resulted in one group being lower in academic ability. The average in the four core subjects for the experimental group was 65.3% while the average for the control group was 65.2% indicating that the groups were not markedly different in academic ability.

Students were interviewed to determine if there was any difference in their access to computers outside of the school program. Access was defined as having a computer in their home or obtaining access at least once a week by have visiting a friend who owned a computer, by a parent bringing one home from work or by the student visiting a computer store. In the experimental group, thirteen students reported having regular computer access while only seven students in the control group reported such access. This was the only factor identified which may account for the difference in the pretest scores.

The analysis of data was continued in order to test the second null hypothesis; that there would be no significant difference between the experimental group and the control group when the pretest and posttest scores were compared. The significance of the difference between the means of the experimental and control group on the grade eight pretest was large enough to result in a statistically significant difference between experimental and control groups for all grades combined. For this reason, analysis of the pretest and posttest results was carried out on a grade by grade basis.

Table 2 is a summary of the mean scores attained by the experimental and control groups at each grade level. All scores are in percentages.

TABLE 2

Mean Scores by Group for Three Grade Levels.

GRADE	GROUP	MEAN SCORES (%)		
		PRETEST	POSTTEST	GAIN
Nine	Experimental	14.83	69.87	55.04
Nine	Control	13.17	14.87	1.70
Eight	Experimental	12.86	59.59	46.73
Eight	Control	5.10	11.05	5.95
Seven	Experimental	8.36	40.55	32.19
Seven	Control	6.42	9.83	3.41

These scores were subjected to an analysis of variance between experimental and control groups using the ANOV26 program on the university computer. A summary of the results obtained appear in table 3.

TABLE 3.

Analysis of Variance on Posttest by Grade.

Gr	Source of Variation	SS	DF	MS	F	Prob.
9	Between Subjects	27661.38	45			
	"A" Main Effects	18454.38	1	18454.38	88.19	0.0000
	Within Groups	9206.94	44	209.25		
8	Between Subjects	24791.96	42			
	"A" Main Effects	17034.91	1	17034.91	90.04	0.0000
	Within Groups	7757.0[illegible]	41	189.20		
7	Between Subjects	5143.31	22			
	"A" Main Effects	3060.73	1	3060.73	30.86	0.0000
	Within Groups	2082.58	21	99.71		

For the grade nine posttest scores, an F-ratio of 88.19 was reported. Using an F critical table, the degrees of freedom reported on the computer printout and a 0.05 level of significance, it was determined that the F critical value needed for rejection of the null hypothesis was 1.63. The null hypothesis was rejected indicating that there was a significant difference between the experimental and control groups at the end of the study. Since the experimental and control groups in grade nine were not significantly different on the pretest, the difference on the posttest may well be due to the treatment.

A similiar analysis of the scores for the grade seven groups indicated an F-ratio of 30.86 and an F critical value of 2.09. On the basis of these figures, the null hypothesis was again rejected. There was a significant difference in the posttest results for grade seven experimental and control groups which may have been attributable to the treatment.

The grade eight analysis of variance indicated an F ratio of 90.04 and an F critical value of 1.69. The null hypothesis was again rejected. While there was a significant difference between the posttest scores of the experimental and control groups, there was also a significant difference between these groups on their pretest scores. Therefore, it cannot be stated with any degree of certainty, that the posttest differences were attributable to the treatment.

Analysis of the results for all three grade levels has indicated that gains achieved by experimental groups were consistently larger than those of the control groups. An examination of the gains achieved by each experimental group was done and Figure 1 depicts the relationship between the mean gains.

Figure 1

Mean Gain Scores for Experimental Groups

```
         60 -!
             !
            -!                     -----
             !                     #####
 Mean    50 -!                     #####
             !                     #####
 Gain       -!            -----    #####
             !            #####    #####
Scores   40 -!            #####    #####
             !            #####    #####
  in        -!            #####    #####
             !            #####    #####
   %     30 -!   -----    #####    #####
             !   #####    #####    #####
            -!   #####    #####    #####
             !   #####    #####    #####
         20 -:----------------------------

                   7        8        9

                       Grade Level
```

This figure indicates that the return on the time and effort invested is relatively low for the grade seven students while the grade eight and nine scores are considerably higher. The posttest mean for the grade seven experimental group was only 40.6% indicating that the course of studies may be too difficult for them.

The grade eight experimental group scores on the posttest had a mean of 60%; just slightly below their overall academic average. Similarily, the grade nine experimental group achieved a mean score on the posttest that is in the range of their academic average with the posttest mean of 69.8% being just above the academic average for the group. This indicates that the course, developed with grade nine students, is appropriate for grade eight and nine students but requires some modification to be used effectively with grade seven students.

The summative evaluation has indicated that the presentation of the course of studies resulted in a significant difference between the experimental group and the control group at all grade levels. However, information obtained from the formative evaluation was of equal importance.

It was determined that a course of studies in computer literacy can be delivered to junior high school students in fifty hours while achieving reasonable sucess in meeting the objectives outlined in the provincial recommendations. An option time block can be used to avoid interfering with the time allocation to the core subjects of Language Arts, Mathematics, Science and Social Studies. A ratio of three students per computer can be used and still give the students a viable "hands-on" introduction to computers. A designated room for a computer laboratory is a definite asset especially in allocating extra computer access time outside of regular classes.

A number of factors were identified which impeded the successful implementation of a computer literacy program and in each case a solution was found. A lack of courseware was overcome by authoring some materials as well as locating and purchasing audio-visual and print materials. A multiple activity group organization, a keyboard orientation component and the allocation of computer access time outside of regular classes made it possible to emphasize a "hands-on" approach with a limited amount of computer hardware. The need for instructors to implement the course of studies was met by offering a thirty hour inservice course to interested staff members as described in a previous chapter.

Informal discussions with students indicated a positive attitude toward computers and the computer literacy course offered. The student demand for access to the Computer Literacy Option still exceeds the ability of the school to meet that demand. In addition, a number of teachers who did not have an opportunity to become involved in the program to date have expressed an interest in taking part in additional inservice training and teaching a computer literacy option if it can be fit into their timetable for next year. It is highly desirable to have extra teachers prepared to present the course of studies to ensure continuity in the presentation of the course of studies in the event that circumstances arise which make it impossible for a teacher to fulfill their committment to the program.

CHAPTER V

It is the intention of this final chapter of this study to summarize the findings, draw conclusions from these findings and make recommendations for further study and action.

Summary and Conclusions.

The problem posed for this problem was; can computer literacy be effectively taught in junior high schools in Alberta within existing option time blocks? Based on the evaluation reported in the preceeding chapter, it can be concluded that computer literacy can be effectively taught as an option in junior high. At all grade levels, the treatment resulted in a statistically significant difference between pretest and posttest scores achieved by the experimental group.

Discussion.

The summative evaluation which has been carried out indicates that scores achieved by the experimental groups are significantly different from the scores achieved by the control groups at each grade level. It can be stated with a reasonable degree of certainty that the difference in scores for grade seven and nine was attributable to the treatment since the groups were equivalent at the beginning of the study. The difference in grade eight scores cannot be attributed to the treatment with the same degree of certainty since a sampling error resulted in a statistically significant difference between the experimental and control groups at the beginning of the study.

An extensive comparison of these two grade eight groups revealed that they were academically equivalent, in the four core subjects, to within 0.1%. However, they differed considerably in their access to computers at the beginning of the study. The experimental group, which had a higher pretest mean, reported that thirteen members had access to computers outside of the school while only seven members of the control group reported similar access. This was the only difference found between the two groups which may account for the difference in performance on the pretest.

The gains from pretest to posttest at each grade level indicated that the older students achieved the greatest gains when taught the coures of studies developed for this study. The mean gain for the grade nine experimental group was 55%, for grade eight it was 47% and in grade seven it was 32%. The posttest mean for the grade seven experimental group was 41% indicating that the course of studies was not well suited to that grade level.

The formative evaluation carried out during this study identified a number of factors which may have contributed to the lower scores achieved by grade seven students. Their overall maturity level may have hampered their progress in that they had some difficulty in adapting to the multiple activity presentation format. They tended to be more unsure of themselves and were slower to function effectively without direct supervision and assistance from the instructor.

It was also noted that a longer period of time was required for grade seven students to attain the same level of keyboarding skills as the eights and nines. There did not appear to be the same difference between the grade eight and nine students. The stage of development of fine muscle coordination may have been a factor in the slower progress with keyboarding. The end result was that the extra keyboarding time used by the grade sevens reduced the time available to teach them computer concepts.

The problem solving skills required to analyse a task and prepare a computer program appeared to be more developed in the grade eight and nine students. This meant that the grade sevens took longer to complete programming tasks which in turn made it difficult to complete the course of studies in the time available.

There appeared to be a greater gap between the grade seven and eight students in all the areas mentioned than was evident between the grade eight and nine students. The instructional team reported that there were no major difficulties in implementing the course of studies with either grade eight or nine students.

Recommendation for Further Study and Action.

The evolving nature of computer literacy and the its inclusion in elementary schools will necessitate ongoing evaluation and upgrading of computer literacy courses for junior high schools. It is recommended that any curriculum developed at the provincial level be reviewed and updated annually.

The attitudes and skills component of computer literacy have not been objectively documented in this study. These components are an important part of computer literacy and studies in these areas should be carried out.

A replication of this study should be done to verify the findings. This would be especially relevant at the grade seven level as the sample of sevens was the smallest. There were a total of forty-six participants at the grade nine level, forty-three at the grade eight level and only twenty-three at the grade seven level. This study was also conducted in the first semester. Grade seven students who had spent five months in a junior high before enrolling in the program may have been able to adapt more effectively to the expectations of a multiple activity class and this should be tested.

The concern for keyboarding skills and habits learned by students at the computer keyboard should be addressed. With students being exposed to computers in elementary and junior high schools, there is ample opportunity to develop habits which may be detrimental to progress in typing skills in high school. The feasibility of offering programs for the introduction of keyboarding to younger students is a subject for further investigation.

The findings of the study indicate that the recommendation of the Computer Literacy Steering Committee that the course of studies be developed for grade eight students is valid. When the gains made by the grade seven and eight students are compared, it appears that the grade eight students derived considerably greater benifit from the course of studies with the same investmemt in time, effort and money. Any course developed for grade eight students would be appropriate for grade nine students with minor modification of some of the assigned tasks. It is recommended that curriculum development continue to focus on the grade eight level as this appears to be the earliest that maximum benifit can be realized from the investment in the program.

A most pressing need for action is in the preparation of teachers to present computer literacy in the schools. A cooperative effort between local school jurisdictions and teacher training institutions will be required to prepare sufficient teachers to meet the rapidly increasing demand for computer literacy programs in the schools. An extensive inservice program will be needed to upgrade teachers presently in the field while the teacher training institutions will be charged with the responsibility of providing new teachers with the necessary expertise to teach computer literacy.

Continued development and testing of a computer literacy curriculum will be required. The importance of ongoing evaluation and upgrading of the curriculum must not be underestimated. The implementation of computer literacy in elementary schools will necessitate modification to the content of junior high school programs. Computer technology is changing so rapidly and will have so profound an impact on society that review of the curriculum less frequently than annually may result in the teaching of irrelevant materials.

The cost and size of computers continues to decrease. This will result in an increasing number of our students at all levels having daily access to these powerful tools. As these devices become as commonplace as calculators, the need for computer literacy courses at the level developed for this study will be reduced and new courses will be required. The need for programming skills may be replaced by the need for an understanding of application of available software. The development of higher level computer languages will simplify the task of harnessing the power of computers for personal benifit. The emphasis in formal education will then shift to an awareness of the range of applications and the skills necessary to benifit from them.

Computer technology will have a far reaching impact on the content and delivery of formal education in the next decade. Educators at all levels will be challenged to keep pace with the rate of change brought about by computers. Failure to change in step with the new technologies will result in the graduation of students who are prepared for the world of yesterday rather than the world of tomorrow.

The course of studies developed for this study should be viewed as one step in an ongoing effort to provide educators and students with tested materials which will assist in the preparation of a computer literate populace which is better prepared for the challenges that they will inevitably face.

7

BIBLIOGRAPHY

Anderson, R.E. National Computer Literacy, 1980. In R.J. Seidel, R.E. Anderson & B. Hunter (Eds.) Computer Literacy. New York: Academic Press, 1982.

Ball, M.J. & Charp, S. Be a Computer Literate. Morristown, N.J.: Creative Computing Press, 1978.

Berger, M. Computers in Your Life. Toronto, Ont.: Fitzhenry & White Ltd., 1981.

Best, J.W. Research in Education. Englewood Cliffs, N.J.: Prentice Hall, Inc., 1977.

Brumbach, K.E. Computer Literacy: 1985. In R.J. Seidel, R.E. Anderson & B. Hunter (Eds.), Computer Literacy. New York: Academic Press, 1982.

Coan. J.S. Basic BASIC: An Introduction to Computer Programming in BASIC Language. (Second Ed.). Rochelle Park, N.J.: Hayden Book Co. Inc., 1978.

Computer Literacy Report and Recommendations. An Unpublished Report submitted to Alberta Education by the Computer Literacy Steering Committee, May, 1982.

Corbett, S. Home Computers: A Simple and Informative Guide. Toronto, Ont.: Little, Brown and Co., 1980.

Covvey, H.D. & McCallister, N.H. Computer Consciousness: Surviving the Automated 80s. Don Mills, Ont.: Addison-Wesley, 1980.

DeRossi, C.J. Learning BASIC Fast. Reston, Virginia: Reston Publishing Co. Inc., 1979.

Englebardt, S.L. Miracle Chip: The Microelectronics Revolution. New York: Lothrop, Lee & Shepard Books, 1979.

Forsythe, K. & Hart, R. Learning in the Eighties. In the Proceedings of The Third Canadian Symposium on Instructional Technology. Vancouver, B.C., 1980.

Gawronski, J.D. Computer Literacy and School Mathematics. The Mathematics Teacher. Vol. 74, No. 8, 1981.

Johnson, D.C., Anderson, R.E., Hansen, T.P., & Klassen, D.L. Computer Literacy - What is it?. The Mathematics Teacher. Vol. 73, No. 2, 1980.

Johnson, M.E. Computer Literacy: What is it?. Business Education Forum. Vol. 35, No. 3, 1980.

Klitzner, C. Coming to Terms with Computer Literacy. Personal Computing. Vol. V, No. 6, 1981

Lamoiter, J.P. Fifty BASIC Exercises. Berkley, California: Sybex, 1981

Leong, W.T. An Investigation of Three Approaches to Teaching Computer Programming to Junior High School Level Students. Unpublished Masters Thesis, University of Toronto, 1975.

Lipscombe, S.D. & Zuanich, M.A. BASIC Fun: Computer Games, Puzzles, and Problems Children Can Write. New York, N.Y.: Avon Books, 1982.

Luehrmann, A. Computer Literacy- A National Crisis and a Solution for it. Byte. Vol. 5, No. 7, 1980.

Moursund, D. Microcomputers Will Not Solve the Computer-in-Education Problem. AEDS Journal. Vol. 13, No. 1, 1979.

Moursund, D. Precollege Computer Literacy: A Personal Computing Approach. Eugene, Oregon. International Council for Computers in Education, Dept. of Computer and Information Sciences, University of Oregon, 1981.

Osborne, A. Running Wild: The Next Industrial Revolution. Berkley, California: Osborne/McGraw Hill Inc., 1979.

Piele, D.T. Micros "GOTO" School. Creative Computing. Vol. 5 No. 9, 1979.

Poole, L. Apple II User's Guide. Berkley, California: Osborne/McGraw Hill inc., 1981.

Solntseff, N. What do we tell Administrators?. Creative Computing. Vol. 7, No. 3, 1981.

Statland, N. Computers- Their Impact on Life Styles in the 1980's. AEDS Monitor. Vol. 17, 1979.

Stevens, D.J. How Educators Perceive Computers in the Classroom. AEDS Journal. Vol. 3, No. 3, 1980.

Watt, D.H. Education for Citizenship in a Computer-Based Society. In R.J. Seidel, R.E. Anderson & B. Hunter (Eds.), Computer Literacy. New York: Academic Press, 1982.

Wilson, J.A.R. Research Guide in Education. Morristown, N.J.: General Learning Press, 1976.

APPENDIX A

OBJECTIVES FOR COURSE OF STUDIES

OBJECTIVES.

TOPIC 1: WHAT IS A COMPUTER

1. Terminology: The students will be able to recognize and use the basic computer terms included in their terminology list handout. This list will include:

   Computer, Disk Drive, Diskette, Keyboard, Video Screen, Hardware, Software, Data, Input, Program, DOS, Debugging, and Cursor.

2. The students will be able to identify five phases in the development of computers.
3. The student will be able to identify the technology associated with each phase. (e.g. mechanical, vacuum tubes, transistors, silicon chips and microelectronics.)
4. The students will be able to state one advantage of any phase over the previous phase.
5. Students will be able to identify the five major parts of a computer system which are common to large or small computer systems.
6. Students will be able to describe the difference between a micro, mini and mainframe computer.
7. The students will be aware of the three basic operations or functions a computer carries out in performing a task and the interrelationship between INPUT, PROCESSING and OUTPUT.
8. The students will recognize that computers are best suited to tasks involving speed, accuracy and the processing of large volumes of information. They will be able to give examples of such tasks.
9. Students will be aware that information is processed inside the computer in the Binary number system which is represented by only ones and zeros.
10. Students will be able to state that ones are represented electrically as power on and zeros as power off.

TOPIC II: HOW TO OPERATE A COMPUTER

1. The student will demonstrate a responsible attitude toward computer equipment and sufficient knowledge to be able to use the available computer system by doing the following:
    - a) Handling floppy diskettes in a manner which will avoid damage and program loss.
    - b) Demonstrating the proper procedure in powering up the microcomputer and "BOOTING" the Disk Operating System. (DOS)
    - c) Using appropriate SYSTEM COMMANDS to enable them to:
        1) Initialize new diskettes using the INIT command.
        2) Produce the listing of programs on a diskette using the CATALOG command.
        3) Load programs from the disk into the computer memory using the LOAD command.
        4) Obtain a listing of a program on the screen using the LIST command.
        5) Run prepared programs using the RUN command.
        6) Save programs onto magnetic storage for permanent storage using the SAVE command.
        7) Lock and unlock programs on a diskette for security against accidental loss using the LOCK and UNLOCK commands.
        8) Remove unwanted programs from a diskette using the DELETE command.
2. The student will be able to identify and correctly use the alphanumeric and special keys on the Apple keyboard.
    - a) The student will use proper finger placement on the alphabetic keyboard according to accepted "touch typing" technique.

  b) The student will demonstrate an understanding of the special function keys on the Apple by by using them appropriately in operating and programming the computer.

3. The students will be able to read and comprehend computer related instructional material, written at an appropriate level, so that they can operate the computer independently.

TOPIC III: COMPUTER PROGRAMMING.

1. The students will recognize that a computer program is the orderly step by step instructions that a computer follows in performing a task.

   a) The steps in a computer program are called statements. Each STATEMENT is made up of a LINE #, COMMAND, and PARAMETERS for that command.

   b) Line numbers determine the sequence in which the steps in a computer program are carried out. Students will demonstrate an awareness of this fact by using an appropriate sequence of line numbers for program steps and by being able to determine what line number should be used to insert an instruction into an existing program.

2. The students will recognize an algorithm as a step by step procedure for solving a problem. They will also recognize that a computer program is a translation of that algorithm into appropriate computer commands. Given a task, the students will be able to:

   a) Outline the procedure required to solve the problem in English statements.

   b) Code the English statements into appropriate computer statements to form a working computer program.

3. The students will recognize the importance of a program to a computer. (analogy to a stereo without a tape or record)

4. The student will recognize that the program stored in the computer memory takes over control of the computer functions when a program is run. This makes it possible for the computer to perform these functions without the operator pressing a button for each function. As a result, the computer is much faster than a calculator for a similar task. The major difference between a calculator and a computer is that the computer can store the program in memory to carry out the functions.

5. Students will recognize that the OUTPUT or end result of a computer program is determined by what is INPUT and how it is PROCESSED. (analogy - baking cookies. If you want to change the output you change the ingredients and how you bake them). Thus, planning a program begins with deciding on the desired OUTPUT.

6. Students will be aware that the computer has commands called IMMEDIATE MODE COMMANDS which do not need a line number in front of them and which will be carried out immediately when the return key is pressed.

   a) The student will be able to demonstrate the use of the following commands in IMEDIATE MODE: HOME, LIST, RUN, PRINT without quotes to do mathematical operations, and PRINT with quotes to display strings.

   b) Students will be aware that some commands can be used in either IMMEDIATE or DEFERRED mode.

7. The students will be aware that DEFERRED MODE COMMANDS are the commands used in programs and that they will always have a line number in front of them so they can be carried out at a later time when the program is run.

8. SCREEN AND PRINTER DISPLAY. The students will recognize that the PRINT COMMAND is used to display output on the screen or printer. They will be able to use the PRINT COMMAND in the following ways:

   a) To display what is enclosed within quotes.

   b) To calculate and display the results of mathematical operations.

   c) To leave a blank line for screen formatting.

   d) To display what is stored as variables.

  e) To display a combination of what is enclosed in quotes and what is stored in variables. (eg. print "Hello ";N$)

9. The students will be aware of the functions of the FLASH, INVERSE, NORMAL and BELL commands and will be able to use them in a program to "dress up" their programs.

1o. SCREEN FORMATTING - the students will be able to plan how they want material to appear on the screen and will prepare programs that use good screen formatting.

  a) Students will use the HOME command to clear the screen and move the cursor to the top left corner of the screen as a first step in formatting a program.

  b) Students will be able to plan the presentation of material using a prepared screen formating sheet.

  c) Students will be able to use the HTAB, VTAB, and TAB commands to position the cursor for screen formatting.

  d) Students will use COMMAS and SEMICOLONS appropriately in print statements to format in columns or leave no spaces as required.

11. VARIABLES - Students will demonstrate an understanding of the nature of variables in the following ways:

  a) By recognizing the analogy between variables stored in computer memory and "mailboxes" with names on them and information stored in them.

  b) By properly naming numeric and string variables using legal names.

  c) Realizing that variables can only store one piece of information at a time so that when new information is put in the old information is discarded.

  d) The students will be able to use the LET command in a program to assign a value or string to a variable.

  e) The students will be able to use the INPUT command in a program to allow the user to enter responses into variables from the Keyboard.

f) The students will be able to use the GET command in a program to accept only one character from the keyboard and then continue without the user pressing the RETURN key.

12. The student will demonstrate an understanding of some of the mathematical capabilities of the computer by being able to:

a) Correctly identify and use the symbols for addition, subtraction, multiplication, division, greater than, less than, equal to, & not equal to.

b) Recognize and use the decision making capability of the computer by utilizing the IF-THEN command in combination with the greater than, less than, and equal to symbols.

c) Use the proper syntax in a computer statement to get the desired order of operation in solving a mathematical problem.

d) Correctly use the random number generator (RND) and integer (INT) commands to be able to produce random whole numbers.

13. The student will demonstrate an understanding of the operation of loops within a computer program by being able to:

a) Create an infinite loop using the GOTO command.

b) Set up an infinite counting loop.

c) Use the IF-THEN statement to control a loop to the desired number of cycles.

d) Use the FOR-NEXT loop in a program to accomplish an assigned task.

e) Use the FOR-NEXT loop as a time delay loop that causes a pause in the program but displays nothing on the screen.

14. Flowcharts. The student will recognize that flowcharts are a graphic representation of the steps in the procedure for solving a problem or performing a task. The students will be able to:

a) Recognizing the standard flowchart symbols for START, END, PROCESS and DECISION.

b) Draw and explain a flowchart for a simple problem.

c) Read a flowchart and write the commands in BASIC which will accomplish the task that is displayed graphically in the flowchart.

15. The student will recognize that computers have graphic capabilities and they will be able to use the appropriate instructions to enable them to:

a) Switch modes between TEXT and GRAPHICS.

b) Set the desired COLOR after referring to the color number chart.

c) Plan figures on the screen formatter sheet and then draw them on the screen in low resolution graphics using the commands GR, COLOR, PLOT, HLIN and VLIN.

16. The students will be able to use the correct syntax of the READ and DATA command combinations to enable them to:

a) Store DATA within a data line so that it can be called up from within a program.

b) Use the READ command to retrieve the desired information from data lines..

17. The students will recognize and use the following characteristics of good computer programs:

a) Adequate documentation to explain the purpose of the program and give instructions for proper use. This information may be either on paper or as part of the program.

b) REM statements will be used within the program to identify various sections or functions.

c) "USER FRIENDLY". The program will adequately explain what is requird for user interaction. User errors anticipated and allowed for in the program.

d) Clear and readable output will be accomplished by proper planning and execution of good screen formatting.

TOPIC 4: COMPUTER USES IN SOCIETY.

1. Students will be able to identify tasks performed by computers in various segments of our society.
2. Students will be able to list uses of microcomputers in the home and in education.
3. Students will recognize that a computer is a piece of electronic hardware that can only do what a program instructs it to do.
4. Students will realize that computer decisions are based on such things as "is A > B " and that they are not capable of making value judgements such as "is George a true friend?"

TOPIC 5: HOW THE COMPUTER AFFECTS SOCIETY.

1. Students will be able to describe ways in which computers can affect their everyday lives.
2. Students will be aware that the impact of computers on society can be positive or negative depending on how humans choose to use them.

APPENDIX B

POSTTEST

COMPUTER LITERACY POSTTEST Name ------------------

Class -----------------

1. Define, in your own words, the following items:

   a)DOS -

   b)CURSOR -

   c)SOFTWARE -

   D)COMPUTER -

2. List the five phases of the developement of computers and indicate the technology associated with that phase.

3. List two advantages that computers in phase four had over computers in previous stages.

4. What are the five major parts of any computer system?

5. What are three differences between a mainframe computer and a microcomputer?

6. List the three basic operations or functions a computer carries out in completing a task?

7. What "number code" does the controller inside a computer work in?

8. How does a computer represent ones and zeros electrically inside the machine?

9. List four points on the proper care and handling of computer diskettes.

10. Fill in the blanks:

    A) ----------- is the command used to get a listing of the programs that are stored on a disk.

B) ----------- is the command used to tell the computer to get a program from the disk and put it into the memory of the computer so it can be used later.

C) ----------- is the command used to tell the computer to carry out the instructions that are stored in its memory.

D) ----------- is the command used to tell the computer to store the program that is in the memory onto the disk.

E) ----------- is the command used to tell the computer to display, on the screen, the lines of a program.

11. Describe in your own words, what a computer program is.

12. What is a STATEMENT in a computer program? Give an example using the proper SYNTAX.

13. The following lines were entered as a new program. What would appear on the screen when the program is RUN?

```
130 END
100 PRINT "TESTING"
120 PRINT "CHECKING"
110 PRINT "EVALUATING"
```

14. What is the main difference between the operation of a computer and the operation of a simple calculator in solving the same problem?

15. In your own words, describe the difference between IMMEDIATE MODE and DEFERRED MODE. Give an example of one Immediate mode command and one Deferred mode command.

16. When the following program is RUN show what will be displayed on the screen and the position in which it will be displayed on on the screen. (as in beside or below the previous information)

```
100 HOME
110 PRINT "BEGINNING A PROGRAM"
120 LET A = 4
130 LET B = 6
140 PRINT A + B
150 LET N$= "FRED"
160 PRINT A
170 PRINT
180 PRINT B
190 PRINT "ALL FINISHED "; N$
200 PRINT THE END
```

17. What is the function of the HOME Command?

18. Write the necessary instructions using HTAB and VTAB to place a "Z" on the screen in a position that is 12 spaces over from the left side of the screen and 8 lines down from the top.

19. Show how the following information will look on the screen when the following program is RUN.

```
100 HOME
110 PRINT 1,2,3,4,5,6
120 PRINT 7;8;9;10;11;12
130 END
```

20. What is the function of the following computer instruction?

```
140 LET B = 27
```

21. For each variable name below, write LEGAL each one that the computer will accept and accept and ILLEGAL beside each name that is not an acceptable variable name.

1) 2A ---------

2) A1 ---------

3) B3$---------

4) #7B ----------

5) ZB ----------

22. What will line 130 print on the screen?

```
100 LET B=4
110 LET B=7
120 LET D=A+E
130 PRINT B
```

23. The LET and INPUT commands both set up a variable and label it. What is the difference between the LET and INPUT commands?

24. What is the computer symbol for the following mathematical functions?

    1) Multiplication ------

    2) Division ------

    3) Greater Than ------

    4) Less Than ------

25. What will the following program display on the screen?

```
100 LET G=10
110 IF G>7 THEN GOTO 130
120 PRINT "TESTING"
130 PRINT "DECISIONS"
140 END
```

26. Line 210 assigns a random number between one and ten to "mailbox" N. Rewrite line 210 in the space below to enable the program to select a number between one and twenty-five.

```
200 FOR B= 1 TO 30
210 LET N= INT(RND(1)*10)+1
220 PRINT N
230 NEXT B
```

27. What will appear on the screen when the following program is RUN.

```
500 PRINT "MOVING ON"
510 PRINT "HERE WE GO"
520 GOTO 500
530 PRINT "TASK COMPLETED"
540 END
```

28. The GET and INPUT commands are used to set up variables, label them, and accept information from the keyboard. What is one way in which these commands are different?

29. What purpose does line 440 serve in the following program?

```
400 IF A=B THEN GOTO 430
410 PRINT "NOT QUITE, TRY AGAIN"
420 GOTO 350
430 PRINT "CORRECT"
440 LET C=C+1
450 GOTO 320
```

30. What is the proper syntax of the IF-THEN command which when placed in line 130 will stop the counting loop when it reaches 25?

```
100 HOME
110 LET W=1
120 PRINT W

130 --------------------------

140 LET W=W+1
150 GOTO 120
160 END
```

31. Using a FOR-NEXT loop, write a program that will print the numbers from 1 to 10 on the screen.

32. What is the purpose of lines 150 and 160 in the following program?

```
120 PRINT "SUPER MATH PROGRAM"
130 PRINT "    WRITTEN BY"
140 PRINT "PROLIFIC PROGRAMMER"
150 FOR D= 1 TO 4000
160 NEXT D
170 HOME
180 PRINT "ARE YOU READY?"
190 INPUT R$
```

7

33. What command is used to to:

a) Switch from normal character display mode to Low Resolution Graphics mode?

b) Switch from the Graphics mode back to the normal character display mode?

34. The READ command is used to set up a "mailbox", place a label on it, and store something in it. Where does the READ command get the information that it stores in the "mailbox"?

35. Give one example of a task that computers perform in each of the following segments of our society.

    a) Government-

    b) Business-

    c) Law Enforcement-

36. Give two uses of microcomputers in each of the following areas.

    a) Home use

    b) Education

37. List three ways that computers may be involved in the daily lives of you and your family.

38. The impact of computers on our society may be either good or bad depending on how they are used. Give one example where you feel that computers can be put to good use and one example where you feel that they may have very undesirable consequences.

APPENDIX C

PRETEST

COMPUTER LITERACY PRETEST.

Name ------------------

Class ----------------

1. Explain, in one or two sentences, what a computer is.

2. List the three functions that are carried out by a computer in completing a task of solving a problem.

3. Both a computer and a simple calculator carry out the same functions in completing a task. What advantage does a computer have over the simple calculator?

4. List two types of tasks that computers are best suited for and give an example of each.

5. What are the five main parts of a computer system? In your own words, describe the function of each part.

6. Give three examples of devices which acan be used to put information into the computer.

7. Give two examples of devices that computers use to inform users of the results of information processing.

8. How are the ones and zeros in the binary code represented electrically inside a computer?

9. Computers have developed through five phases. List the technology or device associated with each phase.

10. List two advantages that computers in phase four had over computers in the previous phases.

11. What "language or code" does the control unit of a computer work in?

12. What is the purpose of the INTERPRETER in a computer?

13. List three languages commonly used by computer programmers.

14. Explain, in your own words, the function of the following keys on the Apple keyboard. (Use an example if you feel it will help you explain better.)

   a) <---
   b) REPT
   c) SHIFT
   d) RESET

15. List four points on the proper care and handling of computer diskettes.

16. How do you find out what programs are stored on a diskette?

17. How would you go about placing a new program on the diskette that you had just finished typing into the computer?

18. How do you erase a program from the diskette?

19. What information is given in the directory listing of the following program?

```
* A 008 AVERAGE
```

20. In your own words, describe what a computer program is.

21. What is a STATEMENT in a computer program? Give an example using proper syntax.

22. In your own words, describe the difference between IMMEDIATE and DEFERRED MODES. Give one example of a command used in each mode.

23. What is the function of each of the following commands?

a) HOME-
b) REM-
c) GOTO 70
d) HTAB 20
e) VTAB 8

24. When the following program is run, what will be displayed on the screen?

```
10 PRINT "THIS IS A TEST"
20 REM OF THE REM STATEMENT
30 PRINT "TEST COMPLETED."
```

25. The computer can store variables in memory "mailboxes" and name these variables for future recall. Beside each variable name given below, state whether the name used is a legal or illegal name for variables.

a) 2A -----------   b) #7B----------
c) A1 -----------   d) ZB ----------
e) B3$-----------

26. What will the following programs display on the screen when they are RUN?

a)

```
100 LET B=7
110 LET B$= "EIGHT"
120 PRINT B,B$
130 END
```

b)

```
100 LET A=1
110 PRINT A
120 LET A=A+3
130 PRINT A
140 END
```

c)

```
100 LET Z$= "GOOD"
110 LET Q$= "MORNING"
120 PRINT Q$;Z$
130 PRINT
140 PRINT "Q$;Z$"
150 END
```

27. What is the instruction on the Apple which will empty the mailboxes in memory so you are sure that there is nothing in them when you start a program?

28. What does line 200 do in the following program?

```
200 LET X= INT(RND(1)*6+1)
210 PRINT X
220 END
```

29. What will appear on the screen when the following program is RUN?

```
100 HOME
110 LET X=1
120 PRINT X
130 LET X=X+1
140 GOTO 120
```

30. The computer is capable of making some simple decisions of a true/false or yes/no type. Give the correct symbol for the following decisions:

a) A is greater than B A ---- B
b) A is less than B A ---- B
c) A is not equal to B A ---- B
d) A is equal to or less than B A ---- B

31. What will the following programs display on the screen?

a)

```
100 LET G= 10
110 IF G > 7 THEN 130
120 PRINT "TESTING"
130 PRINT "DECISIONS"
140 END
```

b)

```
100 LET J=1
110 PRINT J
120 LET J=J+3
130 IF J > 10 THEN 150
140 GOTO 110
150 PRINT" FINISHED"
```

32. Write a computer program using a FOR NEXT LOOP which will act as a time delay as the computer counts from one to one thousand but nothing appears on the screen.

33. Sketch a flowchart of the following program using the appropriate symbols.

```
100 LET A=1
110 PRINT A
120 LET A = A+1
130 IF A = 10 THEN 150
140 GOTO 110
150 END
```

34. Write the necessary instructions to place a number 9 on the screen in a position that is 12 spaces over from the left side of the screen and 8 lines down from the top.

35. What will the following program display on the screen?

```
100 LET K=5
110 FOR Y= 1 TO K STEP 1
120 PRINT Y
130 NEXT Y
140 END
```

36. Write a program that will ask the user what their name is and then store the name the user types on the keyboard in a "mailbox" labelled N$.

37. What would you put in line 40 to cause the computer to tell the user they were right and then end the program when the correct answer is given?

```
10 HOME
20 PRINT "IS 10 * 10 > 100?"
30 INPUT A$
40 ------------------------------
50 GOTO 20
60 PRINT "RIGHT"
70 END
```

APPENDIX D

COURSE OF STUDIES

COMPUTER LITERACY IN JUNIOR HIGH SCHOOLS:

A COURSE OF STUDIES

Prepared by
L.D. Roney
1031 Cantabrian Dr. S.W.
Calgary, Alberta
T2W 1M3

TABLE OF CONTENTS

Objectives

Topic I ---------- 87
Topic II ---------- 88
Topic III ---------- 89
Topic IV ---------- 94
Topic V ---------- 94

SUPPORT MATERIALS

TOPIC I What is a Computer.

Terminology ---------- 95
History of computers ---------- 96
Parts of Computer Systems ---------- 98
Micro, Mini & Mainframe ---------- 101
Limitations of computers ---------- 102
Functions of a computer ---------- 102
What computers do best ---------- 103

TOPIC II How to Operate a Computer

Care of floppy diskettes ---------- 104
Binary code and computer languages ---------- 105
Loading DOS Instructions ---------- 106
Initializing new diskettes ---------- 107
Loading and running programs ---------- 108
Saving programs to diskette ---------- 109
DOS command summary ---------- 110
DOS error messages ---------- 111
Keyboarding (to the teacher) ---------- 112
The Apple Keyboard ---------- 113

TOPIC III Computer Programming

Introduction to programming ---------- 114
Immediate Mode commands ---------- 118
Deferred Mode commands ---------- 118
The PRINT command ---------- 119
Screen Formatting instructions ---------- 120
    HTAB VTAB & TAB ---------- 120
    Comma & Semicolon ---------- 121
    PRINT to leave blank lines ---------- 122

Variables -- 123
    LET command -------------------------------- 124
    INPUT command ------------------------------ 124
    GET command -------------------------------- 124
Math Operations ---------------------------------- 125

Loops -- 126
    Infinite Loops ----------------------------- 126
    Counting Loops ----------------------------- 126
    Loop control with IF-THEN ------------------ 127
    FOR-NEXT Loops ----------------------------- 128
    Time Delay Loops --------------------------- 129
Flowcharting ------------------------------------- 129
Low resolution graphics -------------------------- 131
READ-DATA command -------------------------------- 133
Good computer programs --------------------------- 135
Support materials -------------------------------- 136

TOPIC IV Computer Uses in Society

Support material references ------------------ 137

TOPIC V How Computers Affect Society

Support material references ------------------ 138

OBJECTIVES.

TOPIC 1: WHAT IS A COMPUTER

1. Terminology: The students will be able to recognize and use the basic computer terms included in their terminology list handout. This list will include:

   Computer, Disk Drive, Diskette, Keyboard, Video Screen, Hardware, Software, Data, Input, Program, DOS, Debugging, and Cursor.

2. The students will be able to identify five phases in the development of computers.
3. The student will be able to identify the technology associated with each phase. (e.g. mechanical, vacuum tubes, transistors, silicon chips and microelectronics.)
4. The students will be able to state one advantage of any phase over the previous phase.
5. Students will be able to identify the five major parts of a computer system which are common to large or small computer systems.
6. Students will be able to describe the difference between a micro, mini and mainframe computer.
7. The students will be aware of the three basic operations or functions a computer carries out in performing a task and the interrelationship between INPUT, PROCESSING and OUTPUT.
8. The students will recognize that computers are best suited to tasks involving speed, accuracy and the processing of large volumes of information. They will be able to give examples of such tasks.
9. Students will be aware that information is processed inside the computer in the Binary number system which is represented by only ones and zeros.
10. Students will be able to state that ones are represented electrically as power on and zeros as power off.

TOPIC II: HOW TO OPERATE A COMPUTER

1. The student will demonstrate a responsible attitude toward computer equipment and sufficient knowledge to be able to use the available computer system by doing the following:

    a) Handling floppy diskettes in a manner which will avoid damage and program loss.

    b) Demonstrating the proper procedure in powering up the microcomputer and "BOOTING" the Disk Operating System. (DOS)

    c) Using appropriate SYSTEM COMMANDS to enable them to:

        1) Initialize new diskettes using the INIT command.

        2) Produce the listing of programs on a diskette using the CATALOG command.

        3) Load programs from the disk into the computer memory using the LOAD command.

        4) Obtain a listing of a program on the screen using the LIST command.

        5) Run prepared programs using the RUN command.

        6) Save programs onto magnetic storage for permanent storage using the SAVE command.

        7) Lock and unlock programs on a diskette for security against accidental loss using the LOCK and UNLOCK commands.

        8) Remove unwanted programs from a diskette using the DELETE command.

2. The student will be able to identify and correctly use the alphanumeric and special keys on the Apple keyboard.

    a) The student will use proper finger placement on the alphabetic keyboard according to accepted "touch typing" technique.

b) The student will demonstrate an understanding of the special function keys on the Apple by by using them appropriately in operating and programming the computer.

3. The students will be able to read and comprehend computer related instructional material, written at an appropriate level, so that they can operate the computer independently.

TOPIC III: COMPUTER PROGRAMMING.

1. The students will recognize that a computer program is the orderly step by step instructions that a computer follows in performing a task.

   a) The steps in a computer program are called statements. Each STATEMENT is made up of a LINE #, COMMAND, and PARAMETERS for that command.

   b) Line numbers determine the sequence in which the steps in a computer program are carried out. students will demonstrate an awareness of this fact by using an appropriate sequence of line numbers for program steps and by being able to determine what line number should be used to insert an instruction into an existing program.

2. The students will recognize an algorithm as a step by step procedure for solving a problem. They will also recognize that a computer program is a translation of that algorithm into appropriate computer commands. Given a task, the students will be able to:

   a) Outline the procedure required to solve the problem in English statements.

   b) Code the English statements into appropriate computer statements to form a working computer program.

3. The students will recognize the importance of a program to a computer. (analogy to a stereo without a tape or record)

4. The student will recognize that the program stored in the computer memory takes over control of the computer functions when a program is run. This makes it possible for the computer to perform these functions without the operator pressing a button for each function. As a result, the computer is much faster than a calculator for a similar task. The major difference between a calculator and a computer is that a computer can store the program in memory to carry out the functions.

5. Students will recognize that the OUTPUT or end result of a computer program is determined by what is INPUT and how it is PROCESSED. (analogy - baking cookies. If you want to change the output you change the ingredients and how you bake them). Thus, planning a program begins with deciding on the desired OUTPUT.

6. Students will be aware that the computer has commands called IMMEDIATE MODE COMMANDS which do not need a line number in front of them and which will be carried out immediately when the return key is pressed.

   a) The student will be able to demonstrate the use of the following commands in IMEDIATE MODE: HOME, LIST, RUN, PRINT without quotes to do mathematical operations, and PRINT with quotes to display strings.

   b) Students will be aware that some commands can be used in either IMMEDIATE or DEFERRED mode.

7. The students will be aware that DEFERRED MODE COMMANDS are the commands used in programs and that they will always have a line number in front of them so they can be carried out at a later time when the program is run.

8. SCREEN AND PRINTER DISPLAY. The students will recognize that the PRINT COMMAND is used to display output on the screen or printer. They will be able to use the PRINT COMMAND in the following ways:

   a) To display what is enclosed within quotes.

   b) To calculate and display the results of mathematical operations.

   c) To leave a blank line for screen formatting.

   d) To display what is stored as variables.

e) To display a combination of what is enclosed in quotes and what is stored in variables. (eg. print "Hello ";N$)

9. The students will be aware of the functions of the FLASH, INVERSE, NORMAL and BELL commands and will be able to use them in a program to "dress up" their programs.

1o. SCREEN FORMATTING - the students will be able to plan how they want material to appear on the screen and will prepare programs that use good screen formatting.

a) Students will use the HOME command to clear the screen and move the cursor to the top left corner of the screen as a first step in formatting a program.

b) Students will be able to plan the presentation of material using a prepared screen formatting sheet.

c) Students will be able to use the HTAB, VTAB, and TAB commands to position the cursor for screen formatting.

d) Students will use COMMAS and SEMICOLONS appropriately in print statements to format in columns or leave no spaces as required.

11. VARIABLES - Students will demonstrate an understanding of the nature of variables in the following ways:

a) By recognizing the analogy between variables stored in computer memory and "mailboxes" with names on them and information stored in them.

b) By properly naming numeric and string variables using legal names.

c) Realizing that variables can only store one piece of information at a time so that when new information is put in the old information is discarded.

d) The students will be able to use the LET command in a program to assign a value or string to a variable.

e) The students will be able to use the INPUT command in a program to allow the user to enter responses into variables from the keyboard.

f) The students will be able to use the GET command in a program to accept only one character from the keyboard and then continue without the user pressing the RETURN key.

12. The student will demonstrate an understanding of some of the mathematical capabilities of the computer by being able to:

a) Correctly identify and use the symbols for addition, subtraction, multiplication, division, greater than, less than, equal to, & not equal to.

b) Recognize and use the decision making capability of the computer by utilizing the IF-THEN command in combination with the greater than, less than, and equal to symbols.

c) Use the proper syntax in a computer statement to get the desired order of operation in solving a mathematical problem.

d) Correctly use the random number generator (RND) and integer (INT) commands to be able to produce random whole numbers.

13. The student will demonstrate an understanding of the operation of loops within a computer program by being able to:

a) Create an infinite loop using the GOTO command.

b) Set up an infinite counting loop.

c) Use the IF-THEN statement to control a loop to the desired number of cycles.

d) Use the FOR-NEXT loop in a program to accomplish an assigned task.

e) Use the FOR-NEXT loop as a time delay loop that causes a pause in the program but displays nothing on the screen.

14. Flowcharts. The student will recognize that flowcharts are a graphic representation of the steps in the procedure for solving a problem or performing a task. The students will be able to:

a) Recognizing the standard flowchart symbols for START, END, PROCESS and DECISION.

b) Draw and explain a flowchart for a simple problem.

c) Read a flowchart and write the commands in BASIC which will accomplish the task that is displayed graphically in the flowchart.

15. The student will recognize that computers have graphic capabilities and they will be able to use the appropriate instructions to enable them to:

a) Switch modes between TEXT and GRAPHICS.

b) Set the desired COLOR after referring to the color number chart.

c) Plan figures on the screen formatter sheet and then draw them on the screen in low resolution graphics using the commands GR, COLOR, PLOT, HLIN and VLIN.

16. The students will be able to use the correct syntax of the READ and DATA command combinations to enable them to:

a) Store DATA within a data line so that it can be called up from within a program.

b) Use the READ command to retrieve the desired information from data lines..

17. The students will recognize and use the following characteristics of good computer programs:

a) Adequate documentation to explain the purpose of the program and give instructions for proper use. This information may be either on paper or as part of the program.

b) REM statements will be used within the program to identify various sections or functions.

c) "USER FRIENDLY". The program will adequately explain what is requird for user interaction. User errors anticipated and allowed for in the program.

d) Clear and readable output will be accomplished by proper planning and execution of good screen formatting.

TOPIC 4: COMPUTER USES IN SOCIETY.

1. Students will be able to identify tasks performed by computers in various segments of our society.

2. Students will be able to list uses of microcomputers in the home and in education.

3. Students will recognize that a computer is a piece of electronic hardware that can only do what a program instructs it to do.

4. Students will realize that computer decisions are based on such things as "is A > B " and that they are not capable of making value judgements such as "is George a true friend?"

TOPIC 5: HOW THE COMPUTER AFFECTS SOCIETY.

1. Students will be able to describe ways in which computers can affect their everyday lives.

2. Students will be aware that the impact of computers on society can be positive or negative depending on how humans choose to use them.

TOPIC 1: WHAT IS A COMPUTER

TERMINOLOGY

BASIC - a programming language that uses commands which are similar to English words. The letters stand for (B)eginners (A)ll-purpose (S)ymbolic (I)nstruction (C)ode.

BINARY CODE - The number code made up of ones and zeros. It is used inside all computers because ones can be represented by "power on" and zeros by "power off".

COMPUTER - A high speed information processing device which uses a stored set of instructions to complete a task or solve a problem.

CONTROL UNIT - The part of a computer which directs the flow of information and instructions through a computer.

CURSOR - The flashing square on the screen which indicates where the next character will be displayed.

DATA - The information used by the computer to complete its task. Usually stored in the computer memory temporarily or more permanently on a diskette.

DEBUGGING - The process of discovering and correcting errors in a program.

DISKETTES - (also called Disks or Floppy Disks) a magnetic storage device made up of a magnetic coating on a plastic disk which is enclosed in a black protective envelope.

DOS - Abbreviation for Disk Operating System. These are The instructions which must be loaded into the Apple each time you turn the power on so that the Computer is able to communicate with the disk drive to CATALOG, LOAD and SAVE programs.

HARDWARE - The equipment that makes up the computer system including the central control unit, keyboard, video display screen and the disk drives.

INPUT - Information that is fed into the computer. Usually comes from the keyboard or disk drive.

MEMORY - The part of a computer where programs and data are stored.

OUTPUT - The information which comes out of the computer as the end result or answer to the task it was working on. Usually the output is displayed on the screen or on a printer but may also be saved on a diskette.

PROGRAM - A set of instructions, stored in the computer memory, which the computer will follow to complete a task.

PROGRAMMER - A person who writes programs for a computer.

SOFTWARE - The programs used by the computer to perform a desired task.

HISTORY OF COMPUTERS.

Computers were not invented by any one person following a flash of genius; they evolved gradually over a long period of time. Generally, their development went hand in hand with the search for the solution to problems. Mankind has always attempted to devise tools to make a task faster or easier. The development of devices to make the processing of information faster is no exception.

Progress toward the development of calculating machines was frequently determined by the current state of a technology. This progress can be roughly divided into five phases with each phase being dominated by a particular technology.

PHASE I: THE MECHANICAL ERA. Up to 1943.

During this initial phase the equipment was mechanical in nature and powered by hand. One of the earliest calculating devices was the abacus which was reportedly used by the Chinese in 2600 B.C. In 1642 Blaise Pascal invented the first mechanical adding machine which was hand powered and made up of gears and levers. It wasn't until 1822 that the great grandfather of the computer was dreamed up by Charles Babbage. The machine was a wonder of cogs, gears, sprockets, ratchets and counters. He called it an Analytical Engine and envisioned it solving any type of mathematical problem given to it. It was to include a memory unit within the machine, a conditional transfer device and a memory storage outside the machine. This device was to have functioned much like the computers of today. Unfortunately, the device was never completed because the state of the art of metal working and machine tooling were not advanced enough to build all the intricate parts he needed.

The search for a solution to a problem led to the development of another calculating device. Herman Hollerith was presented with the problem of speeding up the census in the United States. The census was taken every ten years and it looked like it would take ten years to tabulate all the information gathered in the 1890 census. Hollerith used the punched card idea developed for weaving patterns back in the 1800's. Holes were punched in these cards to record census information like age, sex and family size. Then he built a machine with electrified wires to count the data stored on the cards. When the wires encountered a hole in a card, they dropped through to make electrical contact. The signal produced moved a counter around one space. The machine cut the counting time in half. Hollerith went on to set up a company to sell these machines. This company eventually became I.B.M.

For the next fifty years, Hollerith's design was improved but all computers remained basically mechanical until 1943.

PHASE II: THE VACUUM TUBE ERA. 1943 TO MID 1950'S.

In 1943 the University of Pennsylvania and the U.S. Army were trying to find ways to improve the accuracy of artillery shells fired at moving targets such as aircraft. A large number of calculations were needed involving shell speed, target speed, direction and wind. The army was using 100 people with mechanical calculators to work out tables used by gunners to aim the artillery. This was too slow and inaccurate for the task.

Eckert and Mauchly, two scientist from the University, came up with the idea of using vacuum tubes to replace the cogs and gears of the mechanical calculators. When vacuum tube technology was applied to the ideas that Babbage had developed, the first electronic digital computer was invented. It was called the ENIAC or Electronic Numerical Integrator And Calculator. This unit weighed thirty tons and used twenty thousand vacuum tubes. It was also very costly to build and maintain as the tubes burned out frequently. However, it served the purpose for which it was designed and was a significant improvement over the old mechanical calculating devices.

In the late forties and early 1950's computers had advanced to the stage that they were being used by some large corporations but vacuum tube technology required large air-conditioned rooms to house this sensitive equipment to keep it from overheating. The size and cost prevented the majority of the population from accessing computers.

PHASE III: THE TRANSISTOR ERA. MID 5o's to late 196o's.

The transistor was invented in the late 194o's and by the mid 5o's was taking over from the vacuum tube in electronic circuits. The transistor was a replacement for the vacuum tube in computers. It was a fraction the size, used less electricty, gave off less heat, and was much more rugged and reliable than the tube. This made it possible for computers to be built that were much smaller and didn't require a special environment in which to operate. As the cost of production were reduced, computers became practical for more businesses. Previously only governments and extremely large corporations could afford them.

Another advantage of the transistor and associated solid state circuitry was the fact that they could operate so much more rapidly than the old computers with vacuum tubes and electro-magnetic switches. These new computers were capable of millions of calculations per second. This improved speed meant that one main computer could be connected to a number of terminals so that a number of people could be using the computer at the same time with each person using a different program. The computer was so fast that it could keep up with all these users while functioning so rapidly that there was very little waiting time for any of them.

PHASE IV: THE INTEGRATED CIRCUIT ERA. EARLY 197o'S.

The integrated circuit chip first appeared in 1971. This tiny device was another major step forward in making electronic circuits smaller and more reliable. The chip is a tiny wafer of silicon onto which thousands of miniature transistors are etched using a photographic technique. A chip the size of your fingernail contains incredibily complex circuits. They can be mass produced which means that costs are further reduced making the computer accessible to more and more people.

PHASE V: MICROELECTRONIC CONTINUES TO ADVANCE. MID 7o'S to PRESENT.

As the technology of micro-electronics continues to strive for ever smaller, faster and less expensive circuits, computers have advanced also. Perhaps the most significant development in computers in the past ten years has been the microprocessor chip. This integrated circuit contains an entire computer on one chip. A $5 chip today can do the same job as its $50 million dollar vacuum ancestor. With the introduction of these microprocessor chips, various manufacturers began selling small personal computers as kits that could be assembled at home. About 1975 the home computer or personal computer came on the market at a price that made it possible for private individual and small companies to own one.

Each month a new piece of computer equipment or a new add-on device for existing equipment is available. Computers continue to become smaller, faster, less expensive and more powerful. Improved memory systems, better graphic capabilities, music and speech generation units and computers that respond to voice commands are just a few of the ongoing developments in the field of computers.

The above phases in computer development are all part of a continuous growth and development process. The change from one era to another was gradual. Computers built with transistorized circuits were still used for some time after the integrated circuit was invented. For this reason, the dates given for each phase should be considered as approximate.

PARTS OF A COMPUTER SYSTEM

A computer system is made up of five main parts; INPUT, MEMORY, ARITHMETIC AND LOGIC UNIT, CONTROL AND OUTPUT. The CONTROL UNIT, MEMORY and the ARITHMETIC AND LOGIC UNIT are electronic circuits that are housed inside the computer where you don't usually see them or even realize that they are working. The devices that you normally work with are the various INPUT and OUTPUT devices.

INPUT.

There are a number of ways that you can get information into a computer. You can type it in directly from the Keyboard. Magnetic tapes or diskettes can be used to load programs or stored information into the computer. The information from the bubble sheets that you may have used for answer sheets is read into the computer by optical scanners. Some units have light pens which you can point at the screen to feed information into the computer. Other devices that may be used include magnetic ink readers used by banks to read numbers on cheques and punched card readers used by utility companies to send out bills. You can even input information into a computer by using the telephone to connect it with another computer.

MEMORY UNIT.

The memory unit is the part of a computer that stores all the information as it is fed in from the various input devices. The information is stored in separate storage compartments which all have their own 'addresses' so that the computer can find the information very quickly when it is needed. Both the step by step instructions for program and the data that it is working with are stored in the memory unit but in separate parts at different addresses.

ARITHMETIC AND LOGIC UNIT (ALU)

The purpose of this unit is to perform all of the computers mathematical functions and logical decisions. Computers can add, subtract, multiply, divide and compare to see if a number is greater or smaller than another number. Complex calculations are just combinations of basic operations and follow a sequence of stored instructions. Comparing, sorting and rearranging information are the kinds of logical operations done by a computer. Given the proper instructions, a computer can easily sort a list of words alphabetically or arrange a set of numbers in order.

CONTROL UNIT

This unit is the 'traffic director' for the other four parts. When data is fed into the computer, the control unit directs it into the memory unit for storage until it is needed. As a program runs the control unit gets information from memory, sends it to the arithmetic and logic unit for appropriate action, takes the results and directs them back to new locations in memory and finally sends the results to an output device.

OUTPUT DEVICES

These devices are used to display the results in a form that is understandable by you. The most common output device on personal computers like the Apple is the video screen. Signals sent out by the control unit are transformed into letters and numbers on the screen so you can read them. Printers produce the same results on a permanent paper copy. The computer may also send out signals to magnetic storage devices like disks or tapes so that the information can be used later. Signals may be sent to speakers to produce sound or to plotters which may draw a map, graph or electronic circuit.

MICRO'S, MINI'S, AND MAINFRAMES

Computers can be classified into three general categories according to their size, price and capabilities.

While these classifications indicate differences in the computer systems, you will find that all of these systems have many things in common. For example, all of them are made up of the same basic parts (Input, Memory, Control, Arithmetic unit and Output). In each case the systems all work with Binary Numbers inside the machine and use special "Interpreters" to translate this code into a programming language which is easier to understand and use. All of these systems can function with more than one language as long as you have the "interpreter" for that language in the machine.

SUMMARY OF DIFFERENCES

MICROCOMPUTERS	MINICOMPUTERS	MAINFRAME COMPUTERS
COST - thousands	-tens of thousands	- hundreds of thousands
USERS - one at a time	- a few at a time	- many at a time
PROGRAMS - one at a time	- a few at a time	- many at the same time
SPEED OF OPERATIONS - tens of thousands per second	- hundreds of thousands per second	- billions of operations per second
STORAGE - thousands of characters	- millions of characters	- billions of characters

LIMITATIONS OF COMPUTERS

While it is generally agreed that computers are a powerful information processing tool which can handle large quantities of information and process it very quickly, they do have some limitations which have restricted their general usage up to now.

The cost of the computer equipment is one factor which limits the spread of computers. While new technology is reducing the size and cost, it will be some time yet before computers become common in the home.

Secondly, a computer is limited in what it can do. It can be used efficiently in tasks which require repetitive operations on large volumes of information. Many jobs which involve small amount of information that is only used a few times can be handled better manually than by a computer. Often it would take longer to INPUT the necessary information into the computer than it would to do the job with a pen or calculator. In addition, a computer can only do what the program tells it to do and you may not have a program available to do the job you wish done.

A computer is only as good as the quality of program it is given to run. Good computer programs are expensive because it takes many hours to prepare. You may also find that the programs that are available don't do exactly what you need to do so some expert help may be required to change them so they will fit your needs.

Finally, computers and microelectronics are a new technology. Many people are concerned that advances in these areas are proceeding so rapidly that they are getting out of control. Now that computers can store so much information on people, it is necessary to control who has access to that information. Invasion of privacy and computer crimes involving the computers of financial institutions are very real concerns. Until people are more comfortable with this new technology and are convinced that computer abuses can be controlled, they will continue to resist change in this area because of fear of the unknown.

FUNCTIONS OF A COMPUTER

In solving a problem or completing a task, the computer carries out three functions: INPUT, PROCESSING and OUTPUT. The INPUT function takes place as the necessary information is fed into the computer. PROCESSING of the information is carried out according to a set of instructions (or program) stored in the computer's memory. The OUTPUT function is the computers way of reporting the results of processing the information that was input. OUTPUT is most frequently to the TV screen or printer.

While these terms apply to the operation of a computer, they apply equally well to other operations. For example, if you want to bake a batch of cookies, you would INPUT the correct ingredients, PROCESS (measure, mix, and bake) them according to the step by step instructions in the recipe and the OUTPUT would be a batch of perfect cookies. Obviously, if you wanted to bake an Angelfood cake, the INPUT and PROCESSING would be quite different. To obtain the desired OUTPUT you must always use the appropriate INPUT and PROCESSING.

This applies to using a computer as a problem solving tool. If you want the computer to multiply three large numbers together and print out the answer, the three numbers must be INPUT accurately and the computer must have the correct PROCESSING instructions in its memory if the correct OUTPUT or answer is to be obtained.

WHAT COMPUTERS DO BEST

The computer is similar to a calculator in its ability to accept input, process it and display the results. The major difference between the computer and a simple calculator is in the memory. While simple calculators may have a memory which can store values for later use, the processing instructions must be entered by the operator for each problem you wish to solve. The computer has the memory capacity to store both the processing instructions and values. Because the processing steps are stored in memory, the same type of problem can be solved over and over again much more rapidly than you could do them with a calculator.

For example, if you were asked to find the average of your marks this term, you could simply take a calculator and do the calculations. However, if you were asked to calculate the averages for all the students in your school, you could look to a computer to make the job easier. The computer is ideally suited to this task. The steps necessary in performing the calcultion can be stored in the computer memory. As you enter the marks for each student the computer can store them until you have fed in the marks for every student. The instructions that you have placed in the memory can then tell the computer to recall the marks, add them up, do the division, and display the results on the screen or printer. Calculations that would take you hours to do by hand can be done in seconds with the computer. Computers are best at repetitive tasks where accuracy and high speed processing are required.

Computers don't just work with numbers. They can work just as well with letters. Sorting a list of words into alphabetical order is a simple task for a computer and word processor programs make it very easy to type in material or correct errors in your work before the final copy goes onto paper. In general, computers are very good at repetitive tasks where speed and accuracy are important or large amounts of information must be processed.

TOPIC II: HOW TO OPERATE A COMPUTER

CARE OF YOUR FLOPPY DISKETTES

Computer diskettes are made up of a black protective envelope with a small round plastic disk inside. This inside disk has a magnetic coating on it similar to the tape for tape recording. While these diskettes are somewhat flexible, they can be easily damaged by excessive bending.

Diskettes will last for many hours of actual use provided they are taken care of properly. The following rules will help you get maximum life from your diskettes.

1. Always handle diskettes by the black protective envelope. Never touch the grey inner plastic disk.

2. Protect diskettes from extremes of temperature. Proper storage is between 10 and 50 degrees Celcius. Leaving them in the hot sun will cause them to warp just like a record.

3. Don't leave diskettes in the disk drive when the machine is not in use. Prolonged pressure from the drive mechanism may cause them to warp.

4. Keep diskettes away from strong magnetic fields or the information on them may be destroyed. Never leave them on top of the TV.

5. Never write on diskette label with a pencil or ballpoint pen as the pressure may damage the diskette. Use a felt-tip pen or write on the label before placing it on the diskette.

6. Always return diskettes to their paper envelopes when they are not in use. Store them upright in the boxes provided rather than laying them flat on the table or counter where they may be damaged.

7. Don't remove diskettes from the disk drive when the red 'in use' light is on.

8. Always handle diskettes with extreme care as the value of the programs stored on them far exceed the price of the diskette.

BINARY CODE AND THE LANGUAGES USED IN A COMPUTER

A facinating thing about computers is that one machine can work with a number of different languages. The control unit on the computer works in BINARY CODE only (a series of ones and zero's). Each computer must have an 'INTERPRETER' built into it so that inside the computer it can work in BINARY code while it communicates with you in commands that are much easier to understand. When you enter an instruction on the keyboard, the interpreter changes those instructions into number code that the control unit understands. Each key on the keyboard is represented by a standard binary number code that is the same for most computers. This code is called the ASCII code (American Standard Code for Information Interchange). The letter "A" has an ASCII value of 65 or 01000001 in the binary code. The reason that binary code is used inside a computer is because the ones can be represented by having electricity on and zero can be represented by having the electricity off. Thus any letter or number can be represented by a series of "switches" that are either off or on.

EXAMPLE - 'A' = 65 or 01000001 in Binary.

PLACE VALUE	128	64	32	16	8	4	2	1
NUMBER	0	1	0	0	0	0	0	1
SWITCH	off	on	off	off	off	off	off	on

In this combination of switches you have one 64 plus one 1 for a total of 65. Can you set the switches for 'B' which is 66?

The language you will be using on the APPLE is called BASIC (Beginners All-purpose Symbolic Instruction Code). This language is easy to learn because many of the commands are familiar English words. These commands include words like PRINT, RUN, GET, LET, GOTO, IF, THEN, STOP, END. There are many other languages available for computers. Each one has special applications and has advantages over other languages for that application. Some of these languages include FORTRAN, COBOL, PASCAL and LOGO. These languages require a different set of instructions and the computer must have the correct interpreter to change these new instructions into the BINARY CODE. Microcomputers like the Apple can run these languages if you add the necessary 'INTERPRETER' into the machine.

OPERATING THE APPLE II PLUS WITH A DISK DRIVE.

The Apple has built in instructions for loading and saving programs on a tape recorder but cannot work with the disk drive until you load the special Disk Operating System (DOS) instructions into its memory. These instructions must be loaded in each time you turn the power on and they will remain loaded until you turn the power off.

YOU MUST LOAD THE DOS EACH TIME YOU TURN THE COMPUTER ON!

Loading the Disk Operating System (DOS)

1) Insert the diskette into the disk drive. Hold the disk with the label up and your right thumb on the label.

2) Close the door on the disk drive.

** 3) Turn the computer power switch on.

4) Wait until the red 'IN USE' light goes off. (approx. 2o seconds.)

5) If the 'IN USE' doesn't go out and the drive continues to run, hold the CTRL key down and press the RESET key. Remove the disk to see that it is right way up. Repeat steps one through four. If you are not successful on the second try, ask you instructor for assistance.

** if the power has already been turned on:
  - you could turn the power off then on again but that causes unnecessary wear on the switch.
  - instead type PR#6 and press RETURN. This will activate the disk just like turning on the power switch.

One quick test to be sure that the DOS is loaded is to type the command CATALOG. If DOS is loaded the disk drive will be turned on to give you a listing of the programs on the diskette. You will get a syntax error message on the screen if DOS is not loaded.

SETTING UP A NEW DISKETTE (INITIALIZING A DISKETTE).

When you purchase a new diskette it does not contain the necesssary information to load the DOS into the computer. Those instructions must be placed on the diskette so that it can be used later to load, save and run your programs.

STEPS IN INITIALIZING A NEW DISKETTE.

1. Obtain a SYSTEM MASTER DISKETTE, place it in the disk drive and close the door.
2. Type PR#6 and press RETURN. This will load the Disk Operating System instructions into the computer's memory.
3. REMOVE THE MASTER DISKETTE AND INSERT YOUR DISKETTE.
4. Type NEW and press RETURN. This will clear out any old programs that may still be in the memory. (NEW does not erase the Disk Operating System instructions.)
5. Type HOME and press RETURN. This will clear the screen and move the cursor to the top left corner of the screen.
6. Type in the following program. Remember to press RETURN after each line.

```
10 HOME
20 PRINT "PROPERTY OF YOUR NAME"
30 PRINT
40 PRINT "DISK # 1
50 END
```

7. Type INIT HELLO,V1 and press RETURN. The number after the 'V' should be the same as the number in line 40.

8. The red IN USE light will come on and the drive will run for about one minute. During this time the Disk Operating System instructions will be transferred from the computer memory to the diskette and as well your HELLO program will be saved which will identify it as your diskette.

CAUTION!!!!!!!
The INIT command is used only ONCE on a diskette. This command places the disk operating instructions on the diskette, saves the HELLO program and then goes through the disk and ERASES EVERYTHING ELSE OFF THE DISKETTE. You will lose all your programs if you use the INIT command on anything but a blank diskette.

DISK OPERATING SYSTEM COMMANDS

Once the Disk Operating System (DOS) has been loaded into the computer, it is ready to use the diskette as a memory device for permanent storage of programs and information.

LOADING AND RUNNING A PROGRAM WITH THE DISK DRIVE.

1. Type CATALOG and press RETURN to see a listing of the programs on the diskette.

2. Select the program you want. Note carefully the spelling and spaces in the title. (Example- we will choose MATH DRILL)

3. Type RUN MATH DRILL and press RETURN. The red 'IN USE' light will come on and the drive will run briefly while the program is read from the disk and transferred to the computer memory. Once the program is loaded it will begin to run immediately.

4. IF YOU GET A 'BEEP' and a FILE NOT FOUND message you have likely typed in the name differently than it is listed in the CATALOG. No problem -- simply try again.

5. If you wish to load the program without running it immediately, then you would type LOAD MATH DRILL and press RETURN. The program will load from the diskette into the computer memory, stop and wait for further instructions. (you could now LIST or RUN the program)

6. LOADing and RUNning programs that have a "B" in front of them when you CATALOG. Depending on what a file contains, it may run on its own if you type BRUN as the command rather than RUN. Not all files with a B In front can be BRUN but you won't hurt anything by trying. The computer will tell you if it is unable to run it.

SAVING A PROGRAM ONTO A DISKETTE

After you have written a new program or made changes to an old one you will want to save it so that you can use it again. The following steps will save it for you:

1. Choose a name for your program. (eg. COUNTING BY TWO)

2. Type SAVE COUNTING BY TWO and press RETURN. The red 'IN USE' light will come on and the disk drive will run.

3. Wait until the IN USE light on the disk drive goes out. The program has now been transferred from the computer memory to a permanent record on the diskette which will not be erased each time the power is turned off.
   THE PROGRAM HAS BEEN SAVED.

4. CAUTION --- BE SURE THAT YOU DON'T USE THE SAME PROGRAM NAME TWICE by accident. If you already have a program called COUNTING BY TWO and you use the name again, you will erase the old program from the diskette and save the new one over top of it. (This can be useful if you have made changes to a program and you want to erase the old version by saving the new version over top of it.).

   TO GUARD AGAINST ACCIDENTAL ERASING OF PROGRAMS -- You can LOCK your programs so that you will receive a FILE LOCKED message if you try to save a program by the same name over top of it.

5. To LOCK a file - type LOCK COUNTING BY TWO and

press RETURN. The drive will run and an asterisk will be placed in front of the directory listing when you CATALOG.

6. To UNLOCK a file so that you can save a new version over it simply type UNLOCK COUNTING BY TWO and press RETURN.

7. To remove a program from the diskette when you no longer need it -- type DELETE COUNTING BY TWO and press RETURN. The drive will run and the program will be removed. (If the program is locked you will get a FILE LOCKED message. You will have to unlock the file and then delete it.

8. Rules for program names.

   a) Maximum length of the name is 30 characters.

   b) Names must begin with a letter. Numbers and other characters can be used as long as the first character is a letter.

   c) Keep names short and descriptive of program so that you can find the one you are looking for easily when you want to use it later.

DOS COMMAND SUMMARY

1. LOAD

   LOAD HANGMAN - will search the disk for a program named HANGMAN and load it from the disk into the computer memory.

2. RUN

   RUN HANGMAN - will locate the program HANGMAN on the disk, load it into the computer memory and run the program immediately.

3. CATALOG

   This command gives you a listing on the screen of all the prorams on a disk. If there is more than one screen full of titles, the listing will stop while you read it. You can get the next screen full by pressing any key to continue the listing.

4. SAVE

SAVE MATH DRILL - This command will create a new file on the disk, name it MATH DRILL and place whatever program is in the memory of the computer on the disk so that you can load it and use it later.

5. LOCK

LOCK MATH DRILL - protects the program so that you cannot accidently delete it or save another program by the same name over top of it and lose the old program. (You can still LOAD or RUN a locked program.)

6. UNLOCK.

UNLOCK MATH DRILL - unlocks the file so that you can make changes to the program and save the new version in the same disk space under the same name. You could also delete a program after you unlocked it.

7. DELETE.

DELETE MATH DRILL - removes the file from the disk when you no longer want it and makes the space available for other programs. The title is also removed from the directory.

8. DOS ERROR MESSAGES.

a) FILE NOT FOUND - usually this message appears as a result of not typing the file name exactly as it appears in the listing when you CATALOG. (remember spaces count too!)

b) FILE TYPE MISMATCH - this occurs when you try to load or run a file with a T or a B in front of it using the RUN or LOAD commands. Only files with an A or an I in front of them can be run using these commands.

KEYBOARDING

To the teacher - While it is unrealistic to attempt to teach students to type as well as to learn about computers in a semestered option, it is important that a student's first exposure to the keyboard be properly structured. Our experience has been that students who were allowed to develop their own technique at the keyboard, formed habits which made it extremely difficult to develope any degree of speed and accuracy. In addition, the feedback that we were receiving from the high schools was that these students did very poorly in typing classes because of the bad habits that they had learned on the computer keyboard.

A major obstacle to correcting this problem was the lack of training of junior high school teachers in the area of teaching tpying. To overcome this, we purchased an approved typing text and the accompanying set of instructional tapes. The package purchased was PERSONAL APPLICATIONS IN TYPEWRITING by FARMER, GRAHAM, and JENKINS. The text and teachers manual was obtained from the School Book Branch in Edmonton and the tapes from Gage Publishing in Toronto.

Only Component One, Operating the Alphabetic Keyboard, is used in class because of time limitations. A very quick overview of the parts of a typewriter, margin setting and centering are included. The major emphasis is on proper posture, finger positioning and keystroking techniques. In total about eight hours are spent in class time but students are encouraged to borrow the materials and practice outside of the class.

To provide a facility for keyboard orientation, we have obtained a class set of typewriters that have been "pensioned off" within the system. Many of the machines have minor malfunctions which limit their usefulness for production typing but they serve our purposes well.

We highly recommend that some effort at proper keyboarding technique be made for the efficient use of the computers in your classes as well as for the long term benefit of your students.

THE APPLE KEYBOARD.

The apple keyboard is similar to the typewriter keyboard except for the special keys listed below:

POWER

This is not a key but rather an indicator light. The power switch is on the rear of the computer on the left side.

RESET

This key interrupts the computer in the middle of what it is doing and tells the computer to start again. The program is not lost but the run must be started over again. A drastic measure !!! Use only when other methods of interrupting a program will not work.

REPT

This is a rapid repeating key. Holding any key down and pressing REPT will cause that character to print repeatedly until you release the key.

<---

The back arrow is used to correct errors. Each time you press it, the cursor backs up one space so you can correct typing errors.

--->

The forward arrow moves the cursor to the right. It can be used to copy over characters on the screen and has the same effect as retyping those characters.

ESC

The escape key is used in combination with other keys -- not alone. Commonly used to move the cursor around the screen. Pressing the escape key, releasing it, then pressing the 'I' signals the computer to move the cursor up the screen.

CTRL

The control key is similar to the escape key in that it signals the computer that the letter to follow it is a special instruction. The control key is different from the escape key in that it must be held down while you press the letter that it is used with. Example -- Holding the control key while pressing the C will stop the program. Control characters do not show on the screen or printer.

0 (ZERO)

The zero (0) is in the top row of characters next to the nine. When using the computer you must be sure to distinquish between the letter o and a 0 (zero) as the computer sees them as two very different characters.

ONES AND L'S.

On many typewriters the lower case L is used as a one. On the computer, you must use the one if you mean '1' as the computer sees these as different characters.

TOPIC III: COMPUTER PROGRAMMING.

PROGRAMMING

Computer programs are a set of instructions for a computer to follow in completing a task. These instructions must be very exact and in the proper order. Saying to the computer, "You know what I mean" is not a worthwhile effort as the computer will respond with its SYNTAX ERROR message indicating very clearly that it does not know what you mean. Computers only know what you mean if you say it precisely and in the command structure that they understand.

Computer programs are made up of a series of STATEMENTS or LINES of instructions. Each statement must have an exact structure if the computer is going to understand what it is expected to do.

Every statement must begin with a LINE NUMBER. Line numbers determine the order in which the instructions are carried out. It is recommended that you start your programs with line number 100 and increase in steps of 10. (eg. 100 110 120 etc.) This will leave room for you to add line in between your original lines if you forgot something. For example, if you needed to add an instruction between 110 and 120 you could number it 115. Computer line numbers on the APPLE must be whole numbers. Decimal values like 100.8 are not accepted.

Each line number must be followed by a command and then parameters for that command (if they are required).

Examples.

```
LINE#  COMMAND   PARAMETERS


  100  PRINT     "TESTING"

  110  LET        B=42
```

Some commands do not require parameters. These commands are made up of just a line number and a command.

Examples

```
  100   HOME

  110   END
```

The computer carries out the instructions according to the line numbers regardless of the order that they are typed into the computer. Even if you typed in the last line number first, the computer would rearrange the line in order by line number. So if you forget a step, you can always add it later and the computer will insert it into the program in its proper place according to the line number you gave it.

To perform a task efficiently, we must organize the task into a series of steps. In many cases, we have performed the steps so often that we don't even stop to think about them anymore. Lets take the example of getting ready for school in the morning.

1. Turn off the alarm.
2. Get out of bed.
3. Get washed up.
4. Get dressed.
5. Have breakfast.
6. Brush teeth.
7. Put on jacket.
8. Leave for school.

What we have just written is an ALGORITHM or set of steps carried out in performing the task of getting ready for school. If these steps were randomly rearranged as to the order that you did them in, some rather humerous situations could result.

While we wouldn't write a computer program to get ready for school in the morning, we do use the same idea for planning and preparing computer programs. Lets write an ALGORITHM for calculating your average for the four subjects this term.

1. Get Math mark
2. Get Social Studies mark
3. Get Language Arts mark
4. Get Science mark
5. Add all the marks together
6. Divide the total by four
7. Record your average.

To write a computer program to find your average this term, we simply translate each step in the appropriate computer language. In the programming language BASIC, you would need the following instructions.

```
100 INPUT"MATH MARK IS ";MA
110 INPUT"SOCIAL MARK IS ";SS
120 INPUT"LANGUAGE MARK IS";LA
130 INPUT"SCIENCE MARK IS";SC
140 LET T=MA+SS+LA+SC
150 LET A= T/4
160 PRINT "TERM AVERAGE IS ";A
```

When this file is stored in the memory of the computer, it can be run to calculate your term average. The computer operates more quickly than a calculator because all the steps needed to solve the problem are stored in the computer and you don't have to press buttons to tell the computer when to add or divide by 4. All you have to do is enter the numbers for each mark when the computer asks for them. If you were finding the term averages for all the students in your school, you would find it much faster to use a computer than to do all the calculations on a calculator. Doing a large number of term averages is the kind of repetitive task that computers are very good at.

Programs are very important to a computer. Without them, computers are just a complicated piece of electronic circuitry. In order to perform useful functions, a computer needs the precise instructions provided by the program. In this regard, computers are much like a stereo system. The equipment may be attractive and expensive, but without a good quality recording there will be no music coming from it. If you have a poor quality tape you can expect to get poor sound from the stereo.

Similarily, you need good quality programs for the computer if you are to obtain top performance from it. A poorly structured program will either be difficult to work with or you will get inaccurate results. The OUTPUT on both the computer and the stereo are only as good as the INPUT that is fed into them.

The three functions in computer operations are commonly referred to as INPUT, PROCESSING and OUTPUT. In the program for averaging your term marks, the INPUT is the marks for each of you subjects that you typed in using the keyboard. The PROCESSING is what the computer does when it adds the marks together and divides by four. OUTPUT is the answer that is printed on the screen.

There are many instances where these three functions occur besides in the computer. In a stereo the INPUT is the magnetic signals sent in from the tape. The PROCESSING is the changing of these magnetic signals to electricity and sending it to the correct channels for the speakers. The OUTPUT is the sound that you listen to.

When you make a telephone call you also perform these three functions. The INPUT is the numbers that you dial, the PROCESSING is the switching done at the exchange to get the correct line and the OUTPUT is the ringing of your friends phone to signal that you wish to speak to them.

Similarily, you use these functions when you operate a coin vending machine. The INPUT is the coins you put in and the button that you select. The PROCESSING is the activating of the correct mechanism to deliver the correct selection according to the item chosen. Finally, the OUTPUT is the delivery of the item you chose from the machine.

In all of these examples, you needed to know the desired output before you started. In the stereo, you had decided what type of music you wished to hear before you inserted the tape. In the telephone call you had decided which friend you wished to speak to before you dialed the number. And finally, you had decided what item you wanted from the vending machine before you pressed the button.

Once the desired OUPUT has been decided, you can go ahead and plan the INPUT and PROCESSING that will be needed to achieve the desired end result. It is like preparing a desert for lunch. You first decide what you would like (OUTPUT) and then you INPUT the ingredients and PROCESS (mix and bake) them to create the desert you wanted. Desert could be quite a surprise if you begin inputting and processing a bunch of ingredients before you have decided what it is that you wish to make.

Remember this when you begin to write computer programs. You should have clearly determined what the OUTPUT should be before you begin planning the necessary INPUT and PROCESSING steps to achieve that end result.

IMMEDIATE MODE COMMANDS.

An IMMEDIATE MODE COMMAND is one which can be entered into the computer WITHOUT a LINE NUMBER in front of it. These commands are carried out as soon as the RETURN KEY is pressed. All the SYSTEM COMMANDS like NEW, LIST, RUN, LOAD, SAVE, and CATALOG are IMMEDIATE MODE COMMANDS which do not require a line number in front of them. The PRINT command can be used in IMMEDIATE MODE to enable you to use the computer like a calculator.

Example:

`PRINT 4+7` followed by a RETURN will display an 11 on the screen.

These IMMEDIATE MODE calculations can be simple or rather complex like:

```
PRINT 3 * 7 + 4 / 2 - 3
```

Try this on the computer. If you didn't get what you expected, check the ORDER OF OPERATION in the mathematical capabilities section.

The HOME command can be used in IMMEDIATE MODE to clear the screen and move the cursor to the top left corner of the screen.

You will find that a number of COMMANDS will work in either IMMEDIATE MODE or DEFERRED MODE. PRINT, HOME, and LET are examples of these. You will find limited use for IMMEDIATE MODE COMMANDS, except for the SYSTEM COMMANDS, in your work on the computer.

DEFERRED MODE COMMANDS.

These are the commands which are used in programs WITH

LINE NUMBERS in front of them. In many cases they also have PARAMETERS following the command to form the proper SYNTAX of a program STATEMENT. They get their name from the fact that their action is "deferred" or delayed until later when the program is RUN. They are NOT carried out as soon as you press the RETURN.

Any command that is used in a program is being used as a DEFERRED command. The majority of the commands that the computer understands are meant to be used in DEFERRED MODE. Some of these commands that you will use include PRINT, HOME, LET, HTAB, VTAB, END, GOTO, IF-THEN, READ, and DATA.

THE PRINT COMMAND (SCREEN AND PRINTER DISPLAY)

1. The PRINT command is used to display strings on the screen by printing whatever is enclosed between the quotation marks.

```
SYNTAX - 290 PRINT" TYPE IN YOUR CHOICE"
```

2. The PRINT command can be used to calculate and display mathematical operations. (no quotations are required)

```
SYNTAX  200 PRINT 4*5

        210 PRINT (2+1)/3

        220 PRINT A*B
```

Line 200 does the multiplication and prints the answer 20

Line 210 adds the 2+1 first, divides by 3 and then prints the answer 1.

Line 220 multiplies the numbers that are stored in mailboxes A and B. The computer goes to mailbox A and get the number. It then goes to mailbox B, gets the number, multiplies the numbers together and finally display the answer. The only part of this process that you see is the answer on the screen.

3. The PRINT command with nothing behind it (no parameters) can be used to leave a blank line on the screen for screen formatting.

```
SYNTAX 180 PRINT
```

4. The PRINT command can be used to display strings and numerical values stored in variables or "mailboxes".

```
SYNTAX  100 LET A$="GOOD MORNING"
        110 LET B= 7
        120 PRINT A$
        130 PRINT B
```

When this program is RUN the following will appear on the screen:

```
GOOD MORNING
7
```

5. The PRINT command can be used to display a combination of information stored in quotation marks and values stored in variables.

```
SYNTAX  100 N$= "GEORGE"
        110 M = 75
        120 PRINT N$;" YOUR TERM MARK IS ";M
```

When this program is RUN the following will appear on the screen:

```
RUN

GEORGE YOUR TERM MARK IS 75
```

SCREEN FORMATTING INSTRUCTIONS

1. PLANNING SCREEN DISPLAY. You will be given a SCREEN FORMATTING SHEET made up of a 40 by 24 grid to plan your work. This sheet matches the Apple display screen so you can plan how you want the screen to appear before you begin programming.

2. HTAB, VTAB and TAB for screen formatting.

- HTAB can move the cursor to the left or right of its present position to print the next information in the desired column. The range of moves is from column 1 to 40.

```
SYNTAX 100 HTAB 5
       110 PRINT "COLUMN 5"
```

- VTAB can move the cursor either up or down from its present position to print information in the desired line. Range of moves from 1 to 24.

```
SYNTAX  200 VTAB 8
        210 PRINT "LINE 8"
```

- TAB is used inside the print statement to move the cursor across the screen. The TAB command can only move the cursor to the right of its present position just like the tab on a typewriter. TAB is frequently used to print material in columns.

```
SYNTAX  100 PRINT "NAME"; TAB(18) ;"ADDRESS"
```

When this program is RUN it will print NAME at the left side of the screen then move over to column 18 and start printing ADDRESS

Rules 1. TAB can only move cursor to right.
2. The column number must be enclosed in parenthesis - TAB (24)
3. TAB only works horizontally - not vertically.

3. COMMAS and SEMICOLONS as formatting istructions in the PRINT STATEMENT. COMMAS can be used in a PRINT statement as an automatic tab to columns 17 and 33.

```
SYNTAX PRINT"THREE","BLIND","MICE"

       RUN

THREE           BLIND           MICE
```

The T of the word three will be in column 1, the B in blind will be in column 17 and the M in mice will be in column 33.

SEMICOLONS can be used in the PRINT statement to print right next to the previous material without leaving any spaces.

```
SYNTAX 100 PRINT "THREE";"BLIND";"MICE"

       RUN

       THREEBLINDMICE
```

The SEMICOLON can also be used to place material side by side that is entered on different PRINT lines. A SEMICOLON at the end of a line cancels the normal movement of the cursor to the next line when you press the RETURN.

```
SYNTAX  100 PRINT"THREE";
        120 PRINT"BLIND";
        130 PRINT"MICE"

        RUN

        THREEBLINDMICE
```

Rule: If you want a space between words that are separated by SEMICOLONS, you must put the spaces beside the words inside the quotation marks.

4. The PRINT command can be used to leave a blank line for formatting.

```
SYNTAX  100 PRINT "THREE"
        110 PRINT
        120 PRINT "COINS"

   RUN

   THREE

   COINS
```

To leave more than one blank line, you can place a number of PRINT COMMANDS on the same line if you separate them with COLONS (:)

```
SYNTAX  280 PRINT:PRINT:PRINT
```

This instruction will leave three blank lines.

VARIABLES (MAILBOXES)

1. VARIABLES are labelled storage locations in the computer memory. Each variable is given a name so that it can be located for later use. Variables can store either numbers or strings (words) as long as you tell the computer which to expect.

2. The computer will recognize variable names that have one character (Q), two characters (G3), or three characters only if the third character indicates the type of variable (B4$). The $ indicates to the computer that letters are included in the informtion to be stored in the variable.

Rules for variable names -

a) First character must be a letter.

b) The second character (if used) may be either a letter or a number.

c) The last character is recognized only if it indicates the type of variable.

d) You may use words as variable names so that it is easier to understand your programs BUT the computer will only recognize the first two characters and ignore the rest. (If you name it MARK$ the computer calls it MA$)

Legal names	Illegal names
A	7
R3	6K
Tx$	#7M

3. Variables only store one piece of information at a time. You can put new information into the mailbox at any time but when the new information is put in, the old information is discarded and the new is stored in its place.

- You can readily demonstrate this in immediate mode using the LET and PRINT commands. Try these and see.

```
LET B=7
PRINT B
LET B= 3
PRINT B
```

4. The LET command is used to set up variables in memory and place information into them.

```
SYNTAX  100 LET A= 2
        110 LET N$= "FRED"
```

NOTE - STRING or WORD VARIABLES have quotes around them.

5. The INPUT command is also used to set up variables. It differs from the LET command in that INPUT gets its information from the keyboard rather than from within the program line.

```
SYNTAX  100 PRINT " WHAT IS YOUR NAME?"

        110 INPUT N$
```

When these two instructions are RUN the question is printed on the screen by line 100.

Line 110 does the following:

a) Sets aside a "mailbox"

b) Labels it N$

c) Places a question mark (?) on the screen.

d) Waits while you type in your name and press RETURN.

e) Stores what you typed in the "mailbox N$.

6. The GET command is similar to the INPUT except that:

- It only accepts ONE character for storage in the variable.
- It is NOT necessary to press the RETURN after responding to a GET command. As soon as the user presses a key, the computer immediately proceeds to the next instruction.
- Applications - Can be used for any user input where you want to limit it to one letter. Use caution with this command because it doesn't give the user an opportunity to correct typing errors.

  Can be useful in stopping a program while the user reads information from the screen and then presses a key to continue the program.

```
SYNTAX  390 PRINT" PRESS ANY KEY TO CONTINUE..."

        400 GET Q$
```

MATH OPERATIONS

1.Mathematical Symbols on the Apple Keyboard.

+	ADD
-	SUBTRACT
*	MULTIPLY
/	DIVIDE
>	GREATER THAN
<	LESS THAN
=	EQUAL TO
<>	NOT EQUAL TO

2. Order of operations.

The order of operations on the Apple is the same as standard mathematics.

a) Operations in parenthesis are carried out first.
b) Multiplication and division before addition and subtraction
c) Execution progresses from left to right after the above are carried out.

3. The RANDOM NUMBER generator on the Apple will produce numbers between zero and one as a nine digit place decimal number.(eg. .095463857)

To generate whole numbers in a desired range you multiply by a whole number and use the INT command to drop all numbers to the right of the decimal.

```
SYNTAX 200 LET X= INT(RND(1)*10)+1
       210 PRINT X
```

This will generate random numbers between one and ten. To get numbers between one and twenty-five, you would change line 200 so that you multiply by 25 rather than by 10.

LOOPS

1. INFINITE LOOPS USING THE GOTO COMMAND. A computer program can be instructed to stop carrying out instructions in order of the line numbers and jump to another line number to carry on from there. These jumps can be made to a larger line number to skip a section of the program or back to a smaller line number to repeat a section of the program.

   An INFINITE LOOP is a repeating section of a program which has no way of stopping until you interrupt it from the keyboard.

```
SYNTAX 500 PRINT "TEST OF THE GOTO COMMAND."

       510 GOTO 500
```

When this program is RUN it will continue to fill the screen with line after line of:

```
TEST OF THE GOTO COMMAND
```

until you stop it.

An INFINITE LOOP can be stopped by

- Holding CTRL key while you press RESET.
- Holding CTRL key while you press the C key.
- Turning the power off.

2. The COUNTING LOOP is a modification of the infinite loop and is used to keep track of how many times a loop has been repeated.

```
SYNTAX  100 LET M=0

        110 LET M= M+1

        120 PRINT M

        130 GOTO 110
```

-Line 100 sets up mailbox labelled M and stores a zero in it.

-Line 110 reads "Let the new M become the old M plus one". After line 110 is run each time, 1 has been added to whatever was previously stored in the mailbox M.

-Line 120 prints the present value stored in mailbox M.

-Line 130 sends the computer back to line 110 where one more is added to the present value of M.

As this program runs, it will count by ones indefinitely unless you stop it from the keyboard.

3. Bringing the LOOP UNDER CONTROL with the IF-THEN COMMAND. The IF-THEN command allows the computer to make a decision. It causes the computer to jump to another section of the program but only if certain conditions are met. (OFTEN CALLED CONDITIONAL BRANCHING).

SYNTAX

```
140 IF Q=10 THEN GOTO 300

150 GOTO 110
```

Analogy - A "bridge" is needed to get from the IF portion of the command to the THEN portion. Only "truth" can build that bridge.

When the IF portion is true the bridge is strong and you can cross to the THEN side.

When the IF portion is not true, you will "fall through" to the next program line.

In the example given, when Q is equal to 10 the program will branch to line 300. For any value except 10, the program will "fall through" to line 150.

Assignments.

a) Insert the IF THEN COMMAND into a counting loop so that the computer will only count to 100 and then stop.

b) Use the IF THEN to evaluate a string- like IF D$= "YES" THEN GOTO 360.

Note - the YES must be enclosed in quotation marks.

4. The FOR NEXT LOOP is another method of repeating an operation a specified number of times.

```
SYNTAX - 100 FOR B= 1 TO 5 STEP 1

         110 PRINT B

         120 NEXT B

         130 END
```

- Line 100 tells the computer the following:

  a) set up mailbox and label it B

  b) put a 1 in the box as the STARTing number

  c) set the end of the loop when B is greater than 5

  d) increase the value of B in steps of 1 each time through the loop

- Line 120 sends the computer back to line 100 to increase the value of B by one then check to see if B is greater than 5. If B is greater than 5 the computer jumps past the NEXT command to the next larger line number (line 130 end).

TWO PROGRAMS TO DO THE SAME JOB.

```
100 FOR S=1 TO 5 STEP 1      100 LET S=0

110 PRINT S                  110 LET S=S+1

120 NEXT S                   120 IF S>5 THEN GOTO 150

130 END                      130 PRINT S

                             140 GOTO 110

                             150 END
```

NOTE - To have these programs count by two instead of one:

```
100 FOR S=1 TO 5 STEP 2
120 LET S=S+2
```

Note 2. The STEP portion of the for command is optional if you wish to use steps of one. If you just type FOR S=1 TO 5 the computer will automatically assume steps of one.

5. TIME DELAY LOOP. The FOR NEXT loop can be used as a time delay to cause a pause in the program while the user reads instructions on the screen.

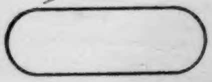
```
SYNTAX  300 FOR T= 1 TO 4000

        310 NEXT T
```

Note- this example will provide about a five second delay. The computer takes about 1 second to go through 800 cycles of the FOR NEXT loop.

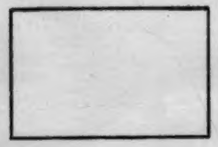
FLOWCHARTING

Flowcharts are picture summaries of the steps carried out in solving a problem or performing a task. They can be used to assist in planning a program or a documentation for your programs so that others can check your flowchart to see how your program operates.

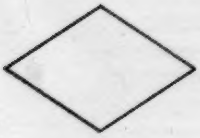
Professional programmers use a large number of different symbols for flowcharting as you can see by looking at flowchart symbol templates. However, we will limit our use to the following symbols:



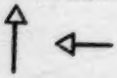
-the terminal symbol used for either the START or the END of a program.



-the PROCESS symbol used to contain an instruction as to what is to be done.



-the DECISION symbol used where a comparison or decision is made. (is A$= "NO" or is B > 7)



-FLOW LINES with arrow heads on them are used to indicate the order of operations direction of flow in a flowchart.

RULES FOR FLOWCHARTING.

1. Always use START and END symbols.
2. Work from top to bottom on the paper.
3. Keep chart neat by leaving suffcient room between symbols and not having flow lines cross.
4. Flow lines should be drawn with a straight edge and should be either vertical or horizontal
5. Be sure to place arrow heads on the flow lines.

ASSIGNMENTS

1. Draw a flowchart which will show the steps taken by a computer in storing two numbers in "mailboxes", adding the two numbers together, and displaying the answer on the screen.
2. Draw a flowchart which will show the steps taken by the computer in asking for three numbers to be typed in then calculating the average for those numbers and displaying the answer on the screen.
3. Draw a flowchart which will show how the computer can place a "secret word" in a mailbox and then have the user guess words until they guess the secret word. The computer will tell them when they get it right.
4. Draw a flowchart of a number guessing game played by two people. Player 1 hides a number and player 2 tries to guess it. After each guess the computer tells the player if the

guess was too high or too low. When the correct number is guessed a message appears on the screen to tell the players the number was guessed.

5. Same as #4 except that a counting loop has been added to keep track of how many guesses before the correct number was found.
6. Translate the flowchart symbols in #5 into program STATEMENTS in BASIC to create a program which will complete the task in #5.

LOW RESOLUTION GRAPHICS.

In addition to displaying letters and numbers on the screen, the Apple can also produce pictures in colored blocks. By telling the computer to switch from TEXT mode to GRAPHICS mode you can have the computer display colored bars or individual blocks through the use of appropriate instructions.

GR - is the command used to tell the computer that you want pictures rather than words.

This command switches the display to a 40 by 40 grid for you to draw on. It also leaves room for four lines of text at the bottom of the screen for printing. Refer to the graphics formatting sheet for an indication of how the display will be set up.

When the GR command is given, the COLOR is automatically set to black and the entire screen is drawn black for a black background. You must select some other color when you wish to draw something since drawing with black on black doesn't show up very well

TEXT - is the command used to cancel graphics and go back to the regular display mode of numbers and letters.

COLOR =7

This command sets the color of the next blocks drawn to a light blue. Any number between 0 and 15 can be used after the equal sign. The color can be changed at any point in the program by inserting a new COLOR command. The Color values are

0	Black	8	Brown
1	Magenta	9	Orange
2	Dark blue	10	Grey
3	Purple	11	Pink
4	Dark green	12	Green
5	Grey	13	Yellow
6	Medium blue	14	Aqua
7	Light blue	15	White

PLOT 20,30

This command is used to plot one individual block. The first number (20) is the column # or how far across the grid. The second number (30) is the row or how far down. In this example a block is plotted in column 20 and row 30.

HLIN - This command is used to draw a horizontal line or bar of color on the screen. The SYNTAX of this instruction is:

```
HLIN  10 , 20 AT  30  <--Draw in row 30

          ------------ End at column 20

       -----------------Start in column 10
```

VLIN - This command is similar to the one above except that it draws VERTICAL bars of color rather than horizontal bars. The SYNTAX for this command is:

```
VLIN 6 , 14 AT  12   <---Draw in column 12

        ----------------End in row 14

     -------------------Start in row 6
```

PUTTING IT ALL TOGETHER.

The following program will demonstrate the commands that you will use in drawing figures in LOW RESOLUTION GRAPHICS MODE.

```
100 Gr                 -Set Graphics mode

110 COLOR =14          -Set color to Aqua

120 PLOT 12,20         -Block in column 12 and row 20
```

```
130 COLOR = 9          -Set color to Orange

140 HLIN 1,25 AT 30    -Bar from column 1 to 25 in row 30

150 COLOR = 6          -Set color to Medium blue

160 VLIN 26,40 AT 40   -Bar from row 26 to 40 in column 40

170 END
```

After running a graphics program, you may wish to return to TEXT mode to list a program or the catalog. You can do this by either using the CTRL-RESET or by typing TEXT and pressing RETURN.

The TEXT command can also be used in the program just before the END to return you to the normal text display mode.

READ and DATA COMMANDS.

The READ command is similar to the LET and INPUT commands in that all three of them are intended to set up variables or "mailboxes" in which information can be stored. The READ command differs from the others in that its source of information is a DATA statement. The information that is to be placed in the mailboxes is picked up from a DATA LINE by the READ command. READ and DATA always work as a team.

```
SYNTAX 190 READ N$
       200 READ M
```

The DATA statement is a line in the computer program which is used to store a string of information which will be needed when the program is run. This saves you having to enter this information from the keyboard each time you use th program. The syntax to a DATA line is as follows:

```
1000 DATA BILL,64,JOHN,35,CANDICE,83
```

Notice, there is no punctuation after the word DATA and there is no puctuation at the end of the line. Each piece of information that is entered must be separated by a comma(,). In addition, NEVER LEAVE ANY SPACES BETWEEN ENTRIES IN THE DATA LINE.

A program can have more than one DATA line. All the data lines will be read as the information that they contain is needed. Reading of the data will always start at the beginning of the data line with the lowest line number and proceed to each higher line number. DATA lines can come before or after the READ command without affecting the operation of the program. However, DATA lines are frequently placed at the end of a program so that they are easy to locate when data changes are necessary.

AS the READ command uses information from the DATA line, a "pointer" is moved along to keep track of which data has already been used and which piece of data is to be read next. If you try to read more data than is contained in the DATA statements in the program, you will get an "out of data" error message and the program will stop.

You must be very careful in organizing the DATA line so that the READ commands find the information in the order that they need it. Missing out one piece of information will cause all the others after that to be read into the wrong memory locations.

EXAMPLE OF READ - DATA PROGRAM.

```
    100 HOME
    110 READ N$
    115 IF N$="ZZZ" THEN GOTO 170
    120 READ A
    130 READ B
    140 READ C
    140 PRINT " TERM MARKS FOR ";N$
    150 PRINT A,B,C
    160 GOTO 110
    170 PRINT " TASK COMPLETED"
    180 END

    1000 DATA BILL,40,65,76,JOAN,70,56,67
    1010 DATA CAROL,80,73,65,FRED,90,87,82,ZZZ
```

When this program is RUN, the four READ statements will pick up the name and marks for each student. The print commands will display these on the screen.

The ZZZ at the end of line 1010 is a "flag" which has been used to signal the computer that the last set of scores has been read. Line 115 checks for the "flag" each time through and ends the program when it finds it. Without the flag and the IF-THEN STATEMENT in line 115 you would get an error message when you ran out of data. You may choose eiher numbers or letters as flags as long as the flag you choose will not appear in the data otherwise. Just be sure to match the IF-THEN decision line to the flag.

GOOD COMPUTER PROGRAMS.

As you become better at writing computer programs, you will be expected to have your programs do more than simply find the answer and print a number on the screen. You should be looking for the following characteristics of good computer programs in your work:

1. The title and opening instructions should explain clearly the purpose of the program and how it is to be used. If the instructions are complex, they should be available on paper so the user can refer to them as the program is in use.

2. The program is well planned and organized with the various sections being clearly identified by REM STATEMENTS.

3. The presentation of information and results should be well formatted so that it is logically organized and easily read.

4. The program should be "USER FRIENDLY". The instructions should be clear enough so that anyone can understand how to use it. In addition, you should try to anticipate errors that are likely to be made by a user and build in ways of letting them try again rather than having to start all over again. Remember, good instructions are an important part of a "user friendly" program.

You have written a good program if you can bring any student in and have them run the program successfully without any prompting from you.

TOPIC III SUPPORT MATERIALS.

1. Sound filmstrip series- BASIC: An Introduction to Computer Programming.

    a) Part 1- BASIC is a Language. PRINT, LET, NEW, RUN and LIST are introduced.

    b) Part 2- Introduction to Computer Programming. Deals with Looping using GOTO, IF-THEN, FOR-NEXT, BREAK and INPUT.

    c) Part 3- Problem solving strategies. Includes identifying key info in a problem, Flowcharting, and writing programs.

    d) Part 4- Loop logic patterns for more complex programs. A continuation of part 3.

2. Be a Computer Literate by Ball and Charp.

    a) Chapter 5. Communicating with a computer.

    b) Chapter 7. How to write a simple program.

3. BASIC Fun: Computer Games, Puzzeles, and Problems Children Can Write. by Lipscomb and Zuanich

    Excellent source of program tasks for various commands at a level all beginners can understand. Problem, sample run and program listing given for each task.

TOPIC IV SUPPORT MATERIALS.

1. Be a Computer Literate by Ball, M. and Charp, S.
    a) Chapter 8. How computers work for us.
2. Computers in Your Life by Berger, M. Select chapters from table of contents to deal with topic of your assignment
3. Home Computers by Corbett, S.
    a) Chapter 2- Computers We Already Live With.
4. Sound Filmstrips: Computers in Our Society by Encyclopedia Britannica Educational Corp.
    a) Computers in Everyday Life.
    b) Computers in School and Business.
    c) Personal Computers in the Home.

TOPIC V SUPPORT MATERIALS.

1. Home Computers by Corbett, S.
    - a) Chapter 1. Computers for Everyone.
    - b) Chapter 9. Computer Crime and Abuse.
    - c) Chapter 10. The Computerized Home.
2. Sound filmstrip. Computers in our Society by Encyclopedia Britannica Educational Corp.
    - a) Computers Change Society.
    - b) Computer Careers.